THE DINI GROUP

LOGIC Emulation Source

User Guide
DN6000K10

LOGIC EMULATION SOURCE

DN6000K10 User Manual Version 1.1

© The Dini Group

1010 Pearl Street • Suite 6
La Jolla, CA92037
Phone 858.454.3419 • Fax 858.454.1279
support@dinigroup.com
www.dinigroup.com

Table of Contents

ABOUT THIS MANUAL 1

1 MANUAL CONTENTS 1
2 ADDITIONAL RESOURCES 1
3 CONVENTIONS 2
3.1 Typographical 2
3.2 Online Document 3
4 RELEVANT INFORMATION 4

GETTING STARTED 5

1 PRECAUTION 5
2 THE DN6000K10 LOGIC EMULATION KIT 5
3 INSTALLATION INSTRUCTIONS 7
3.1 Jumper Setup 7
3.2 Jumper Description 8
3.3 Switch Setup and Description 10
3.4 Oscillator Setup 11
3.5 PPC RS232 Port Setup 11
3.6 Powering ON the DN6000K10 11
4 PLAYING WITH YOUR DN6000K10 VIA THE USB INTERFACE 12
5 PLAYING WITH YOUR DN6000K10 VIA THE PPC'S 12

INTRODUCTION TO USB CONTROLLER SOFTWARE 14

1 EXPLORING THE SOFTWARE TOOLS 14
1.1 USBController 14
1.1.1 Getting Started with USBController 15
1.1.2 Basic Menu Operations 15
1.1.3 File Menu 16
1.1.4 Edit Menu 16
1.1.5 FPGA Configuration Menu 16
1.1.6 FPGA MemoryMenu 17
1.1.7 Settings/Info Menu 18

INTRODUCTION TO VIRTEX-II PRO AND ISE 20

2 VIRTEX-II PRO 20
2.1 Summary of Virtex-II Pro Features 20
2.2 PowerPC™ 405 Core 21
2.3 RocketIO 3.125 Gbps Transceivers 21
2.4 Virtex-II FPGA Fabric 22
3 FOUNDATION ISE 6.1I 24
3.1 Foundation Features 24
3.1.1 Design Entry 24
3.1.2 Synthesis 25
3.1.3 Implementation and Configuration 25
3.1.4 Board Level Integration 26
4 VIRTEX-II PRO DEVELOPER'S KIT 26

INTRODUCTION TO THE REFERENCE DESIGN 28

1 EXPLORING THE REFERENCE DESIGN 28
1.1 What is the Reference Design? 28
1.2 Using the Reference Design 29

1.3 Compiling The Reference Design *31*
1.3.1 The Xilinx Embedded Development Kit (EDK) 31
1.3.2 Synplicity Synplify 31
1.3.3 Xilinx ISE 31
1.3.4 The Build Utility: Make.bat 31
2 GETTING MORE INFORMATION 39
2.1 Printed Documentation *39*
2.2 Electronic Documentation *39*
2.3 Online Documentation *39*

PROGRAMMING/CONFIGURING THE HARDWARE 40

1 PROGRAMMING THE CONFIGURATION FPGA 40
2 MCU DETAILS / PROGRAMMING THE MCU 45
3 CONFIGURING HYPERTERMINAL 46
4 CONFIGURING THE FPGA USING SELECTMAP 47
4.1 Bit File Generation for SelectMAP Configuration *47*
4.2 Creating Configuration File "main.txt" *52*
4.2.1 Verbose Level 52
4.2.2 Sanity Check 53
4.2.3 Format of "main.txt" 53
4.3 Starting SelectMAP Configuration *55*
4.3.1 Description of Main Menu Options 56
4.4 Bitstream Encryption *59*

BOARD HARDWARE 60

1 INTRODUCTION TO THE BOARD 60
1.1 DN6000K10 Functionality *61*
2 VIRTEX-II PRO FPGA 62
2.1 FPGA (2VP70) Facts *62*
3 FPGA CONFIGURATION 63
3.1 Micro Controller Unit (MCU) *63*
3.1.1 MCU EEPROM Interface 64
3.1.2 MCU SRAM External 64
3.1.3 MCU FLASH 64
3.1.4 MCU General Purpose IO (GPIO) 65
3.1.5 MCU USB 2.0 Interface 65
3.1.6 RS232 Interface 66
3.2 Configuration FPGA *66*
3.2.1 Configuration PROM/FPGA Programming 68
3.2.2 Design Notes on the Configuration FPGA 68
3.3 SmartMedia *69*
3.3.1 SmartMedia Connector 70
3.3.2 SmartMedia connection to Spartan (Configuration FPGA)/MCU 70
3.4 Boundary-Scan (JTAG, IEEE 1532) Mode *71*
3.4.1 FPGA JTAG Connector 71
3.4.2 FPGA JTAG connection to Configuration FPGA 71
4 CLOCK GENERATION 72
4.1 Clock Methodology *72*
4.2 Clock Source Jumpers *77*
4.2.1 Clock Source Jumper Header 78
4.3 Roboclocks *78*
4.3.1 RoboClock PLL Clock Buffers 78
4.3.2 RoboClock Configuration Jumpers 80
4.3.3 Roboclock Configuration Headers 84
4.3.4 Useful Notes and Hints 84
4.3.5 Customizing the Oscillators 85
4.3.6 Common Clock Source Selections 85
4.4 External Clocks *85*
4.4.1 External SMA Clock 86
4.4.2 Connections between FPGA's and External SMA Clock Buffer 86
4.5 DDR Clocking *87*
4.5.1 Clocking Methodology 87
4.5.2 Connections between FPGA's and DDR PLL Clock Buffer 88
4.6 Power PC (PPC) Clock – Sytem Clock *89*
4.6.1 Clocking Methodology 89

4.6.2 Connections between FPGA's and System Clock Buffer ... 89
4.7 Rocket IO Programmable Clocks ... 90
4.7.1 Clocking Methodology ... 91
4.7.2 ICS8442 Programmable LVDS Clock Synthesizer ... 91
4.7.3 Connections between FPGA's and RocketIO Clock Synthesizers ... 91
5 RESET TOPOLOGY ... 93
5.1 DN6000K10 Reset ... 93
5.2 PPC Reset ... 95
6 MEMORY ... 95
6.1 FLASH ... 95
6.1.1 FLASH Connection to the FPGA's ... 96
6.2 DDR SDRAM ... 103
6.2.1 Basics of DDR Operation ... 104
6.2.2 DDR SDRAM Configuration ... 104
6.2.3 DDR SDRAM Clocking ... 105
6.2.4 DDR SDRAM Termination ... 105
6.2.5 DDR SDRAM Power Supply ... 106
6.2.6 DDR SDRAM Connection to the FPGA ... 107
7 ROCKET IO TRANSCEIVERS ... 124
7.1 SMA Connectors ... 125
7.1.1 FPGA to SMA Connector ... 125
8 CPU DEBUG AND CPU TRACE ... 127
8.1 CPU Debug ... 128
8.1.1 CPU Debug Connectors ... 128
8.1.2 CPU Debug Connection to FPGA's ... 129
8.2 CPU Trace ... 129
8.2.1 CPU Trace Connectors ... 130
8.2.2 Combined CPU Trace/Debug Connection to FPGA's ... 130
9 GPIO LED'S ... 131
9.1 Status Indicators ... 131
9.2 FPGA A GPIO LED's ... 133
10 POWER SYSTEM ... 134
10.1 Stand Alone Operation ... 134
10.1.1 External Power Connector ... 135
10.1.2 Power Monitors ... 136
10.1.3 Power Indicators ... 136
10.1.4 Front Panel Indicator/Switch ... 136
11 TEST HEADER & DAUGHTER CARD CONNECTIONS ... 137
11.1 Test Header ... 137
11.1.1 Test Header Connector ... 139
11.1.2 Test Header Pin Numbering ... 139
11.2 DN3000K10SD Daughter Card ... 140
11.2.1 Daughter Card LED's ... 142
11.2.2 Power Supply ... 143
11.2.3 Unbuffered IO ... 144
11.2.4 Buffered IO ... 144
11.2.5 LVDS IO ... 144
11.2.6 Connection between FPGA and the Daughter Card Headers ... 145
12 MECHANICAL ... 171
12.1.1 Case ... 171
12.1.2 PWB DimensionThe DN6000K10 PWB conforms to the following dimensions: ... 172
12.1.2 The DN6000K10 PWB conforms to the following dimensions: ... 173

APPENDIX A – ADDRESS MAPS ... 174

FPGA A ... 175
FPGA B ... 176
FPGA C ... 177
FPGA D ... 178
FPGA E ... 179
FPGA F ... 180
FPGA G ... 181
FPGA H ... 182
FPGA I ... 183

List of Figures

Figure 1 - DN6000K10 LOGIC Emulation Board 6
Figure 2 - Default Jumper Setup 8
Figure 3 – DN6000k10 Not Found 15
Figure 4 – Configuration PROM/FPGA Programming Header 41
Figure 5 - New Project Screen Shot 48
Figure 6 - Input File 48
Figure 7: New Project Dialog Box 49
Figure 8: Project Navigator 50
Figure 9 - Main Menu 56
Figure 10 - Interactive Configuration Option Menu 58
Figure 11 - DN6000K10 Block Diagram 61
Figure 12 - MCU EEPROM Interface 64
Figure 13 - MCU SRAM 64
Figure 14 - MCU FLASH 65
Figure 15 - MCU General Purpose IO Connector 65
Figure 16 - USB Connector 65
Figure 17 - MCU Serial Port 66
Figure 18 – Configuration PROM/FPGA Programming Header 68
Figure 19 - SmartMedia Connector 70
Figure 20 - FPGA JTAG Connector 71
Figure 21 - Clocking Block Diagram 72
Figure 22 - LVPECL Clock Input and Termination 78
Figure 23 - Clock Source Jumper 78
Figure 24 - RoboClock Functional Block Diagram 80
Figure 25 - RoboClock Configuration Jumpers 84
Figure 26 - External SMA Clock 86
Figure 27 - DDR DCM Implementation 88
Figure 28 - PPC External Clock 89
Figure 29 - REFCLK/BREFCLK Selection Logic 91
Figure 30 - Reset Topology Block Diagram 94
Figure 31 - FLASH Connection 95
Figure 32 - DDR SDRAM Connection 104
Figure 33 - SSTL2 Class 1 Termination 105
Figure 34 - SSTL2 Class 2 Termination 106
Figure 35 - DDR VTT Termination Regulator 107
Figure 36 - RocketIO Block Diagram 125
Figure 37 - CPU Debug Connector 128
Figure 38 - Combined Trace/Debug Connector Pinout 130
Figure 39 - ATX Power Supply 135
Figure 40 - External Power Connection 135
Figure 41 - Optional PWR Connector 136
Figure 42 - Front Panel Indicator/Switch 137
Figure 43 - Test Header 138
Figure 44 - Test Header Pin Numbering 139
Figure 45 - DN3000K10SD Daughter Card Block Diagram 140
Figure 46 - DN3000K10S Daughter Card 141
Figure 47 - Assembly drawing for the DN3000K10SD 142
Figure 48 - PM7200 Server Case 172

List of Tables

Table 1 – Jumper Description 8
Table 2: S2 Dipswitch Configuration Settings 55
Table 3: HyperTerminal Main Menu Options 56
Table 4: HyperTerminal Interactive Configuration Menu Options 59
Table 5 - FPGA Configuration Modes 69
Table 6 - FPGA configuration file sizes 69
Table 7 - Connection between Configuration FPGA/MCU 70
Table 8 - FPGA JTAG connection to Configuration FPGA 72
Table 9 - Clocking inputs to the FPGA's 73
Table 10 - Clock Source Signals 77
Table 11 - RoboClock Configuration Signals 80
Table 12 - Connection between FPGA and External PPC Oscillator 86
Table 13 - Connection between FPGA's and DDR PLL Clock Drivers 88
Table 14 - Connection between FPGA and External PPC Oscillator 89
Table 15 - Connections between FPGA's and Rocket IO Clock Synthesizers 92
Table 16 - PPC Reset 95
Table 17 - Connection between FPGA and FLASH 96
Table 18 - Connection between FPGA's and DDR SDRAM's 107
Table 19 - Connections between FPGA and SMA Connectors 125
Table 20 - RocketIO Performance 127
Table 21 - CPU Debug connection to FPGA 129
Table 22 - Combined CPU Trace/Debug connection to FPGA 130
Table 23 - CPLD LED's 132
Table 24 - MCU LED's 132
Table 25 – FPGA A GPIO LED's 133
Table 26 – Voltage Indicators 136
Table 27 - External Power Connections 143
Table 28 - Connection between FPGA and the Daughter Card Headers 145

Chapter 1

About This Manual

This User Guide accompanies the DN6000K10 LOGIC Emulation Board. For specific information regarding the Virtex-II Pro parts, please reference the datasheet.

1 Manual Contents

This manual contains the following chapters:

Chapter 1, "Getting Started", contains information on the contents of the LOGIC Emulation Kit.

Chapter 2, "Introduction to the Virtex-II and ISE", an overview of the Vitex-II platform and the software features.

Chapter 3, "Introduction to the Software Tools", information regarding the reference design and test software.

Chapter 4, "Programming/Configuring the Hardware", step-by-step information on programming and configuring the hardware.

Chapter 5, "Board Hardware", detailed description of board hardware.

2 Additional Resources

For additional information, go to http://www.dinigroup.com. The following table lists some of the resources you can access from this website. You can also directly access these resources using the provided URLs.

Resource	Description/URL
User Manual	This is the main source of technical information. The manual

Resource	Description/URL
	should contain most of the answers to your questions
Dini Group Web Site	The web page will contain the latest manual, application notes, FAQ, articles, and any device errata and manual addenda. Please visit and bookmark: http://www.dinigroup.com
Data Book	Pages from The Programmable Logic Data Book, which contains device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging http://support.xilinx.com/partinfo/databook.htm
E-Mail	You may direct questions and feedback to the Dini Group using this e-mail address: support@dinigroup.com
Phone Support	Call us at **858.454.3419** during the hours of 8:00am to 5:00pm Pacific Time.
FAQ	The download section of the web page contains a document called **DN6000K10 Frequently Asked Questions (FAQ)**. This document is periodically updated with information that may not be in the User's Manual.

3 Conventions

This document uses the following conventions. An example illustrates each convention.

3.1 Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: -100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild design_name`**
Garamond bold	Commands that you select from a menu	**File →Open**
	Keyboard shortcuts	**Ctrl+C**

Convention	Meaning or Use	Example
Italic font	Variables in a syntax statement for which you must supply values	ngdbuild *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Braces [ ]	An optional entry or parameter. However, in bus specifications, such as bus[7:0], they are required.	ngdbuild [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**lowpwr =**{**on**|**off**}
Vertical bar |	Separates items in a list of choices	**lowpwr =**{**on**|**off**}
Vertical ellipsis . . .	Repetitive material that has been omitted	IOB #1: Name = QOUT' IOB #2: Name = CLKIN' . .
Horizontal ellipsis **. . .**	Repetitive material that has been omitted	**allow block** *block_name* *loc1 loc2 ... locn;*
Prefix "0x" or suffix "h"	Indicates hexadecimal notation	Read from address 0x00110373, returned 4552494h
Letter "#" or "_n"	Signal is active low	INT# is active low fpga_inta_n is active low

3.2 Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example

Blue Text	Cross-reference link to a location in the current file or in another file in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Red Text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Handbook*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest datasheets.

4 Relevant Information

Information about PCI can be obtained from the following sources:

Reference the PCI Special Interest Group for the latest in PCI/PCI-X Specifications:

PCI Special Interest Group http://www.pcisig.com

2575 NE Kathryn St. #17

Hillsboro, OR 97124

FAX: (503) 693-8344

Other recommended specifications include:

PCI Industrial Computer Manufacturers Group (PICMG) http://picmg.org

401 Edgewater Place, Suite 500

Wakefield, MA 01880, USA

TEL: 781-224-1100

FAX: 781-224-1239

Suggested reference books (available from Amazon):

Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis,* ISBN: 0-13-451675-3

Sundar Rajan, *Essential VHDL: RTL Synthesis Done Right*

Edwin Breecher, *The IQ Booster: Improve Your IQ Performance Dramatically*

Chapter 2

Getting Started

Congratulations on your purchase of the DN6000K10 LOGIC Emulation Board! You can begin by installing the software, or by powering on your DN6000K10. If you wish to begin installation, please follow the installation instructions. The remainder of this chapter describes the contents of the box and how to start using the DN6000K10 LOGIC Emulation Board.

1 Precaution

The DN6000K10 is sensitive to static electricity, so treat the PCB accordingly. The target markets for this product are engineers that are familiar with FPGA's and circuit boards, so a lecture in ESD really isn't appropriate (and wouldn't be read anyway). However, the following web page has an excellent tutorial on the "Fundamentals of ESD" for those of you who are new to ESD sensitive products:

http://www.esda.org/basics/part1.cfm

The DN6000K10 has been factory tested and pre-programmed to ensure correct operation. You do not need to alter any jumpers or program anything to see the board work. A reference design is included on the enclosed CD. Please verify that the board is in working order by following the steps below:

2 The DN6000K10 LOGIC Emulation Kit

The DN6000K10 LOGIC Emulation Kit provides a complete development platform for designing and verifying applications based on the Xilinx, Virtex-II Pro FPGA family. The DN6000k10 is stand-alone or hosted via a USB interface. The DN6000K10 enables designers to implement embedded processor based applications with extreme flexibility using IP cores and customized modules. The Virtex-II Pro

FPGA with its integrated PowerPC processor and powerful Rocket I/O, Multi-Gigabit Transceivers (MGT) make it possible to develop highly flexible and high-speed serial transceiver applications.

The DN6000K10, in its standard configuration, includes a high speed USB interface, a SmartMedia interface for configuration, 16M x 16 DDR SDRAM (x8), 4M x 16 FLASH (x5), RS232 ports (x4 multiplexed) and a RS232 monitor port. There are 9 low skew clock sources that are distributed to the FPGAs and the test header. A 200-pin test header allows for connection to individual FPGA's IO banks, using a custom daughter card. Figure 1 - shows the DN6000K10 Logic Emulation Board.

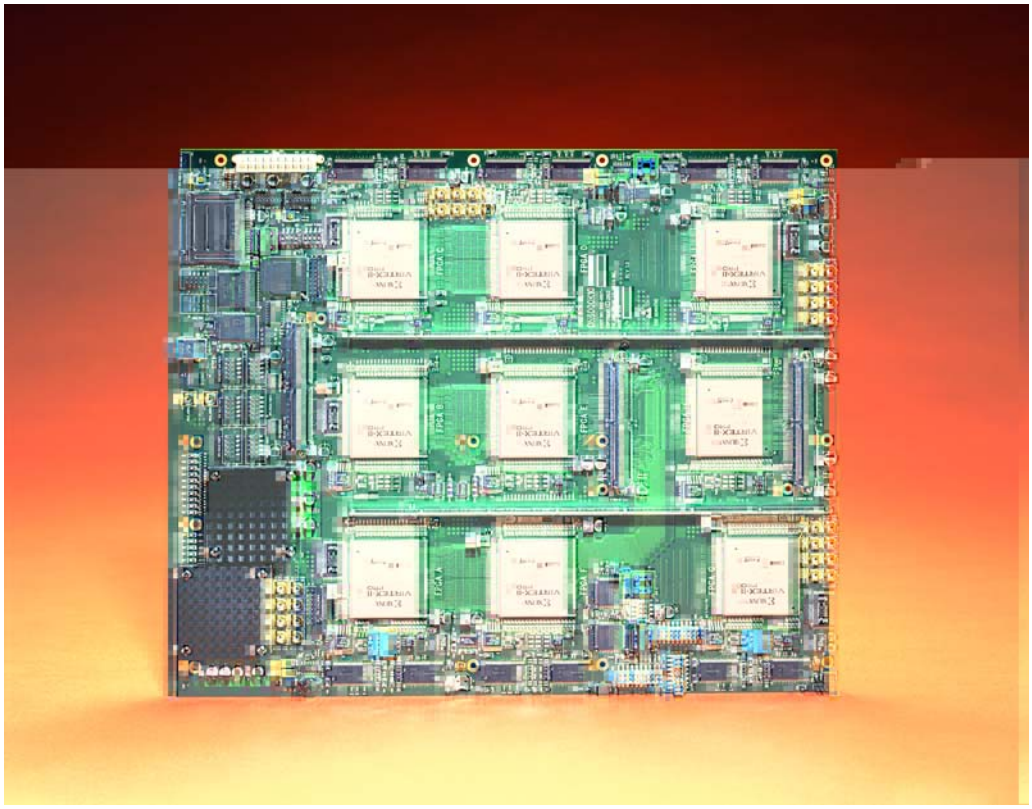


Figure 1 - DN6000K10 LOGIC Emulation Board

The DN6000K10 LOGIC Emulation Kit includes the following:

- ✓ DN6000K10 development board (2VP70 or 2VP100 in the FF1704 package) Note: Specific speed grade parts required for various RocketIO/Power PC operating speeds, refer to Xilinx datasheet).
- ✓ 32MB SmartMedia Card, with reference design and main.txt
- ✓ 32MB SmartMedia Card, for customer use (blank)

- ✓ FlashPath Adapter to copy bit files to the SmartMedia Card(s)
- ✓ RS232 Serial cable, female to female (6ft)
- ✓ IDC 10-pin to DB 9-pin adaptor cable
- ✓ Jumpers 0.1"(x10)
- ✓ Documentation/Reference CD

Optional items that support development efforts (not provided):

- ✓ Xilinx ISE software
- ✓ JTAG cable
- ✓ Daughter Card

3 Installation Instructions

3.1 Jumper Setup

Figure 2 indicates the factory jumper configuration of the DN6000K10.

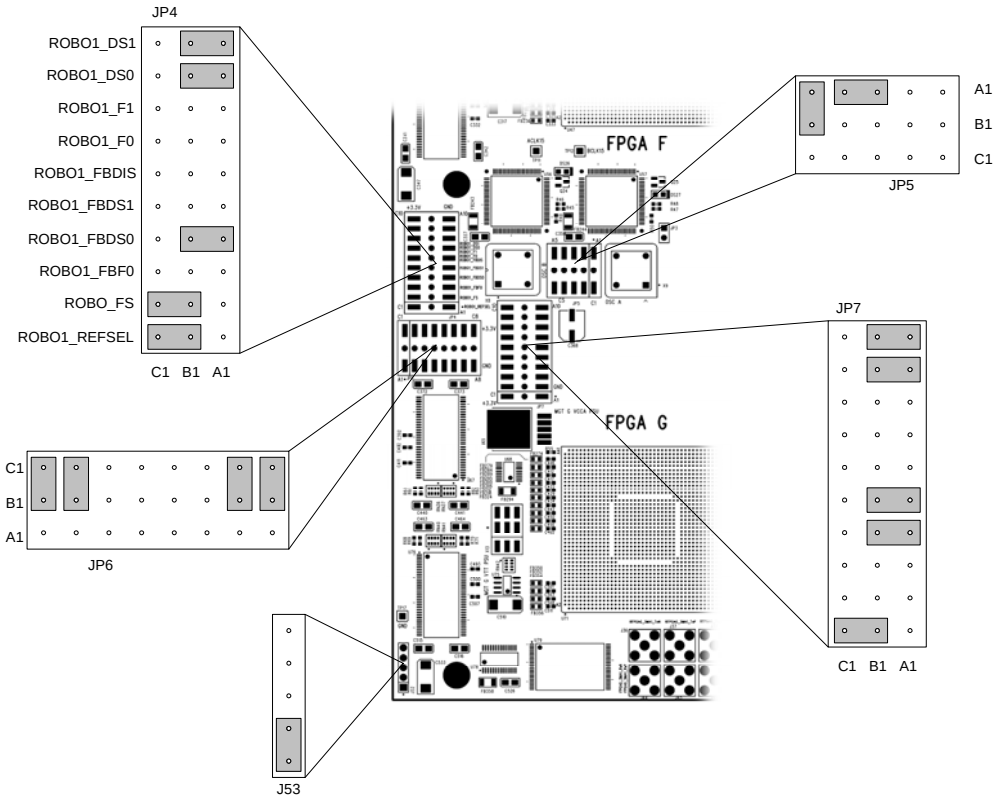


Figure 2 - Default Jumper Setup

3.2 Jumper Description

Table 1 – describes the functionality of the installed jumpers on the DN6000K10.

Table 1 – Jumper Description

Jumper Installed	Signal Name	Description
JP5.A4-A3	PLL2BNC	CFPGA_CLKOUT connected to RoboClock #2 (U57). CFPGA_CLKOUT is an output clock from the Configuration FPGA. This connection causes 48MHz to output on all BCLK signals, which is used in the reference design for communication between the Configuration FPGA and VirtexII Pro FPGAs.
JP5.A5-B5	CLOCKB	Oscillator (X8 – 33.33MHz) connected to RoboClock #1 (U56).
JP4.A1-C1	ROBO1_REFSEL	ROBOCLOCK #1, Reference Select Input: The REFSEL input controls how the reference input

Jumper Installed	Signal Name	Description
		is configured. When LOW, it will use the REFA pair (PLL1A) as the reference input. When HIGH, it will use the REFB pair (PLL1BC, PLL1BNC) as the reference input. This input has an internal pull-down.
JP4.A2-C2	ROBO1_FS	ROBOCLOCK #1, Frequency Select: This input must be set according to the nominal frequency (fNOM). Refer to Table 1 in the datasheet.
JP4.A4-B4	ROBO1_FBDS0	ROBOCLOCK #1, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.
JP4.A9-B9	ROBO1_DS0	ROBOCLOCK #1, Output Divider Function Select: Controls the divider function of all banks (ACLKx) of outputs. Refer to Table 4 in the datasheet.
JP4.A10-B10	ROBO1_DS1	ROBOCLOCK #1, Output Divider Function Select: Controls the divider function of all banks (ACLKx) of outputs. Refer to Table 4 in the datasheet.
		RoboClock #2 (U57)
JP7.A1-B1	ROBO2_REFSEL	ROBOCLOCK #2, Reference Select Input: The REFSEL input controls how the reference input is configured. When LOW, it will use the REFA pair (PLL1A) as the reference input. When HIGH, it will use the REFB pair (PLL1BC, PLL1BNC) as the reference input. This input has an internal pull-down.
JP7.A4-B4	ROBO2_FBDS0	ROBOCLOCK #2, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.
JP7.A5-B5	ROBO2_FBDS1	ROBOCLOCK #2, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.
JP7.A9-B9	ROBO2_DS0	ROBOCLOCK #2, Output Divider Function

Jumper Installed	Signal Name	Description
		Select: Controls the divider function of all banks (BCLKx) of outputs. Refer to Table 4 in the datasheet.
JP7.A10-B10	ROBO2_DS1	ROBOCLOCK #2, Output Divider Function Select: Controls the divider function of all banks (BCLKx) of outputs. Refer to Table 4 in the datasheet.
JP6.B1-C1	OSCA	Enable for Oscillator A (X9)
JP6.B2-BC2	OSCB	Enable for Oscillator B (X8)
JP6.B7-C7	ROBO1_MODE	ROBOCLOCK #1, Output Mode: This pin determines the clock outputs' disable state. When this input is HIGH, the clock outputs will disable to high-impedance (HI-Z). When this input is LOW, the clock outputs will disable to "HOLD-OFF" mode. When in MID, the device will enter factory test mode.
JP6.B8-C8	ROBO2_MODE	ROBOCLOCK #2, Output Mode: This pin determines the clock outputs' disable state. When this input is HIGH, the clock outputs will disable to high-impedance (HI-Z). When this input is LOW, the clock outputs will disable to "HOLD-OFF" mode. When in MID, the device will enter factory test mode.
		MISC
J53.1-2	PS_Ons	Power jumper.

3.3 Switch Setup and Description

Switch	Default Position	Signal Name	Description
S2.1	On	FPGA_MSEL0	FPGA MSEL[0] – used to set configuration mode for all VirtexII Pro FPGAs
S2.2	Off	FPGA_MSEL1	FPGA MSEL[1] – used to set configuration mode for all VirtexII Pro

Switch	Default Position	Signal Name	Description
			FPGAs
S2.3	Off	FPGA_MSEL2	FPGA MSEL[2] – used to set configuration mode for all VirtexII Pro FPGAs
S2.4	Off	DP_SW3	Not used
S4.1	Off	CFPGA_MSEL0	Configuration FPGA MSEL[0]
S4.2	Off	CFPGA_MSEL1	Configuration FPGA MSEL[1]
S4.3	Off	CFPGA_MSEL2	Configuration FPGA MSEL[2]
S4.4	Off	Not Connected	N/A

3.4 Oscillator Setup

The DN6000k10 is shipped from the factory with a 33.33MHz in X2, 14.31818 in X3, and 100MHz in X4. If the Roboclock jumpers are set to their default locations then ACLKx will be 133.33MHz and BCLKx will be 48MHz.

3.5 PPC RS232 Port Setup

There are 4 RS232 ports that are shared with the 9 VirtexII FPGAs. These ports are multiplexed by the Configuration FPGA and can be changed via the MCU Main Menu (see Configuring HyperTerminal). The default setup is:

Port1 (P3): FPGA A

Port2 (P4): FPGA F

Port3 (P6): FPGA G

Port4 (P7): FPGA C

3.6 Powering ON the DN6000K10

This section describes what is necessary to power-up the DN6000K10.

1. Install the SmartMedia card containing reference design into the DN6000K10.

2. If switch position 4 on S2 is OFF then the MCU will automatically boot from the flash, and try to configure the FPGAs via the SmartMedia card (please see Creating Configuration File "main.txt" for information on setting up the files

on the SmartMedia card). If switch position 4 on S2 is ON then the MCU will wait for USB commands and will not be able to configure the FPGAs until the USB application (on the product CD in "Source Code\USBController\USBController.exe") is opened.

3. You can hook up the MCU RS232 port P2 to see messages during FPGA configuration (see Configuring HyperTerminal for more details).

4. Plug the ATX power supply into J17 and turn on the power. The power will not turn on unless a jumper is installed in J53.1-2.

4 Playing with your DN6000k10 via the USB interface

At this point, the DN6000k10 should be powered on. All present FPGAs should be programmed with the reference design bit files provided by The Dini Group.

1. Hook up the USB cable to your DN6000k10 and your PC.

2. When you plug in the board and start windows the first time windows should detect the board and ask for a driver. Select "install from a list" -> select "search for the best driver in these locations". Select "include the location in the search" and browse to where the INF file is located (on the product CD in "Source Code\AETEST_USB\driver\win_wdm\") ->select "finish"

3. If the driver was installed successfully you should see the following device in the USB section of the device mananger: "DiniGroup DN6000k10 FLASH boot".

4. You can now run the USB application found on the product CD in "Source Code\USBController\USBController.exe".

5. Please see

5 Playing with your DN6000K10 via the PPC's

At this point, the DN6000K10 should be powered on. All present FPGA's should be programmed with the reference design bit files supplied by The Dini Group.

6. Hook up the PPC RS232 port 1 (P3). All PPC RS232 ports run at 19200 bps.

7. Press '1' on the MCU menu to reconfigure all FPGA's. When configuration is complete the following text will be displayed on the PPC RS232 port:

```
****************************************
****************************************
** DN6000K10 ASIC DEVELOPMENT PLATFORM **
******* REFERENCE DESIGN SOFTWARE *******
****************************************
****************************************

FPGA_A:
Waiting for External Host Commands
Press Any Key To Enter Local User Menu
```

8. At this point tests may be run from the MCU menu. Text will appear on the PPC RS232 port as tests from the MCU menu are run on the associated FPGA (At this point the PPC port is connected to FPGA A).

9. Press a key on the PPC RS232 port to display the PPC test menu. See the section Using the Reference Design in Chapter 4: Introduction to the Reference Design for more information.

Congratulations! You have now programmed the DN6000K10 and successfully executed our utility to exercise various features of the board. All of the source code for the embedded PowerPC utility is included on the CD for reference. The FPGA design, written in Verilog, can also be found on the CD and used as a basis for a new design.

Chapter 3

Introduction to USB Controller Software

1 Exploring the Software Tools

1.1 USBController

USBController application is used to communicate with the DN6000k10.

All USBController source code is included on the CD-ROM shipped with the DN6000k10. USBController can be installed on Windows 98/ME/2000/XP. There is a command line version called AETEST_USB that can be installed on Linux and Solaris. Detailed installation instructions for each version may be found in README on the CD.

There are 2 versions of the USBController

- USBControllerUpdate.exe – Allows the user to update the MCU Flash
- USBController.exe – Does not allow the user to update the MCU Flash

The USBController Application contains the following functionality:

- Configure FPGA(s) over USB
- Verify Configuration Status
- Configure FPGAs via Smartmedia card
- Clear FPGA(s)
- Reset FPGA(s)
- Set RocketIO CLK Frequency

- Turn FPGA Fan(s) Off/On
- Retrieve MCU/Spartan version

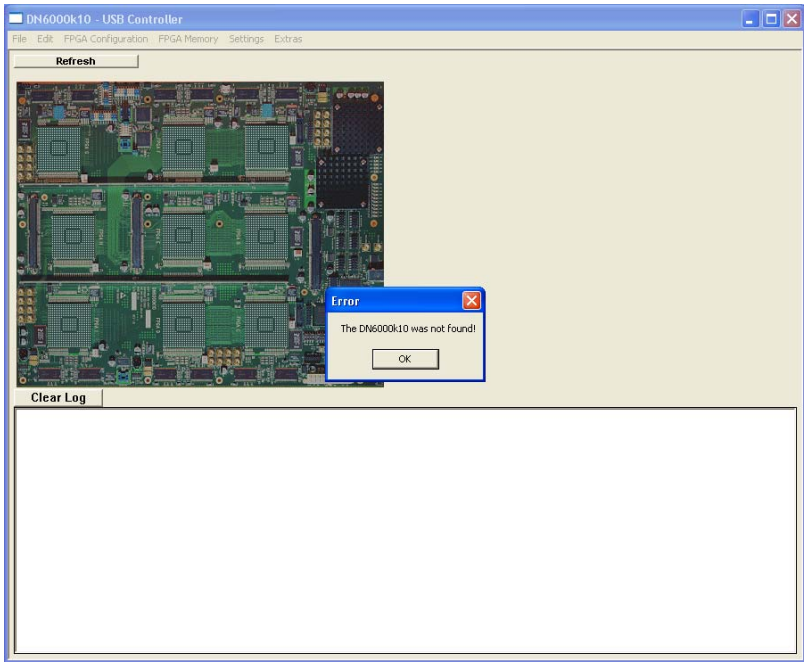
The following features are available when The Dini Group Reference design bit files are loaded:

- Read/Write to FPGA(s) – see Appendix A for address maps
- Test DDRs/FLASH/Reigsters/Interconnect

1.1.1 Getting Started with USBController

Once USBController is installed and the DN6000k10 is powered on and the USB cable is plugged in, the user can open USBController. The USBController application should immediately find the DN6000k10. If USBController does not find the DN6000k10, the user will get the following alert (see Figure 3)

Figure 3 – DN6000k10 Not Found



1.1.2 Basic Menu Operations

If the USBController finds the DN6000k10 and the USB cable was plugged into the PC before power was turned on to the DN6000k10 you will see the following screen:

If the USB Cable was plugged into the DN6000k10 after it powered on you will see the following screen:

1.1.3 Enable/Disable USB to FPGA Communication

This button allows you to disable the USB to FPGA communication via the Spartan II. When the USB interface is used, the Spartan II will drive main bus (MB) pins 0-39 in order to provide USB communication to the FPGAs. This makes main bus pins 0-39 unusable for any other purpose. If your design requires the use of these pins it is necessary to disable USB to FPGA communication, which will cause the Spartan II to cease driving these pins and release them for other purposes.

Note: USB to FPGA communication is disabled by default.

Note: In order to run our reference design, USB to FPGA communication **must** be enabled.

1.1.4 File Menu

The File Menu has the following 2 options:

(1) Open – opens a file with the selected text editor (notepad by default). To change the text editor see Settings/Info Menu section

(2) Exit – Closes the USBController application

1.1.5 Edit Menu

The Edit Menu performs the basic edit commands on the command log in the bottom half of the USBController window. The Find option is not currently supported.

1.1.6 FPGA Configuration Menu

The FPGA Configuration Menu has the following options:

(1) Refresh Configuration Status – Queries to see which FPGA(s) are configured and update the GREEN LEDS in DN6000k10 picture

(2) Configure via USB (individually) – After selecting this option a window will pop and ask which FPGA you want to configure and then what bitfile you want to configure the selected FPGA with. The status of the FPGA configuration will detailed in the log window and the DN6000k10 will be updated after the bitfile has been transferred.

(3) Configure via USB using file – This option allows the user to configure more than one FPGA over USB at a time. To use this option you must

create a setup file that contains information on which FPGA(s) should be configured and what bitfiles should be used for each FPGA. The file should be in the following format, the first character of each line represents which FPGA you want configured (a-i or A-I), this letter should be followed by a colon and then the path to the bitfile to use for this FPGA. The path to the bitfile is realative to the directory where this setup file is, or you can use the full path. Below is an example of an accepted setup file:

A: fpga_a.bit

B: fpga_b.bit

C: fpga_c.it

(4) Configure via SmartMedia Card – This option allows the user to use a SmartMedia card to configure the FPGAs. Please section Creating Configuration File "main.txt" for information on what files should be on the SmartMedia card to use this option.

(5) Clear All FPGAs – This option will clear all FPGAs of configuration.

(6) Reset – This options sends an active low reset (active for approx. 20ns) to all FPGAs on the signal called FPGA_GRSTn which is connect to the following I/O pins:

FPGA A: M28

FPGA B: M22

FPGA C: M22

FPGA D: M21

FPGA E: M16

FPGA F: E20

FPGA G: E20

FPGA H: M28

FPGA I: A26

1.1.7 FPGA Memory Menu

The FPGA Memory Menu has the following options

(1) Write DWORD(s) – Writes DWORD(s) to memory with a specified starting address, and number of DWORD(s) to write. Also, the user can specify what to write. (address value, inverse of address, or a user inputted value) Additionally, enabling verbose mode will allow the user to see what has been written, to what address. Please note that all addresses must be entered as 8-digit hexadecimals.

(2) Read DWORD(s) – Reads DWORD(s) from memory with a specified starting address, and number of DWORD(s) to read. The "Values used to test memory" options are non-functional for this dialog. Verbose mode will print what is read from what address.

(3) Write and Read DWORD(s) – this combines the previous two items. It first writes the DWORD(s) to a given address range, and then reads back those addresses. The (values used to test memory" will determine what is written to each address.

(4) Test Address Space – this will write the specified value to a given address range, read it back, and check the results for errors. Note: some memory addresses cannot be written to, and will return errors. Please check the FPGA memory maps in Appendix A for clarification.

(5) Display Address Space – coming soon!

(6) Test DDR (through PPC's) – tests an FPGA's DDR by using the PPC built into each FPGA.

(7) Test FLASH (through PPC's) – tests an FPGA's flash by using the PPC built into each FPGA.

(8) Test SRAM (through PPC's) – tests an FPGA's SRAM by using the PPC built into each FPGA.

(9) Test Internal Registers – coming soon!

(10) Test Interconnect – coming soon!

(11) Test ALL (through PPC's) – tests an FPGA's DDR, flash, and SRAM by using the PPC built into each FPGA.

(12) Display Memory Map – coming soon!

1.1.8 Settings/Info Menu

The Settings/Info Menu has the following options

(1) Set FPGA RocketIO CLK Frequency – When the DN6000k10 is first powered up the RocketIO CLK inputs to the FPGAs are inactive. The RocketIO CLK Inputs are connected to the following FPGA Differential CLK inputs on all FPGAs: F21/G21 and AT21/AU21. This menu option allows the user to specify what frequency the RocketIO CLKs should be set at for each FPGA. The supported frequency range is 31.25MHz – 700MHz. After selecting this option, a pop-up window will ask which FPGA's RocketIO Frequency you want to set (or you can choose to set all to the same frequency), and then what frequency you want. Check the log window to verify what frequency the CLKs were actually set at.

(2) Change Text Editor – This options allows the user to select a text editor to use (the default editor is notepad).

(3) FPGA Stuffing Information – This option will display the type of FPGAs that are stuffed on the DN6000k10.

(4) Turn FPGA Fans On/Off – This option will either turn the FPGA fans on or off.

(5) MCU Firmware Version – This option will display the MCU Firmware version in the log window.

(6) BOARD/SPARTAN Version – This option will display the Board Version along with the Spartan (Config Fpga) Version.

Chapter 4

Introduction to Virtex-II Pro and ISE

2 Virtex-II Pro

The Virtex-II Pro FPGA solution is the most technically sophisticated silicon and software product development in the history of the programmable logic industry. The goal was to revolutionize system architecture "from the ground up." To achieve that objective, the best circuit engineers and system architects from IBM, Mindspeed, and Xilinx co developed the world's most advanced FPGA silicon product. Leading teams from top embedded systems companies worked together with Xilinx software teams to develop the systems software and IP solutions that enabled new system architecture paradigm.

The result is the first FPGA solution capable of implementing high performance system-on-a-chip designs previously the exclusive domain of custom ASICs, yet with the flexibility and low development cost of programmable logic. The Virtex-II Pro family marks the first paradigm change from programmable logic to programmable systems, with profound implications for leading-edge system architectures in networking applications, deeply embedded systems, and digital signal processing systems. It allows custom user-defined system architectures to be synthesized, next-generation connectivity standards to be seamlessly bridged, and complex hardware and software systems to be co-developed rapidly with in-system debug at system speeds. Together, these capabilities usher in the next programmable logic revolution.

2.1 Summary of Virtex-II Pro Features

The Virtex-II Pro has an impressive collection of both programmable logic and hard IP that has historically been the domain of the ASICs.

- High-performance FPGA solution including:
  - Up to twenty-four RocketIO™ embedded multi-gigabit transceiver blocks (based on Mindspeed's SkyRail™ technology)

  - Up to four IBM® PowerPC™ RISC processor blocks
- Based on Virtex™-II FPGA technology
  - Flexible logic resources, up to 125,136 Logic Cells
  - SRAM-based in-system configuration
  - Active Interconnect™ technology
  - SelectRAM™ memory hierarchy
  - Up to 556 Dedicated 18-bit x 18-bit multiplier blocks
  - High-performance clock management circuitry
  - SelectIO™-Ultra technology
  - Digitally Controlled Impedance (DCI) I/O

2.2 PowerPC™ 405 Core

- Embedded 300+ MHz Harvard architecture core
- Low power consumption: 0.9 mW/MHz
- Five-stage data path pipeline
- Hardware multiply/divide unit
- Thirty-two 32-bit general purpose registers
- 16 KB two-way set-associative instruction cache
- 16 KB two-way set-associative data cache
- Memory Management Unit (MMU)
  - 64-entry unified Translation Look-aside Buffers (TLB)
  - Variable page sizes (1 KB to 16 MB)
- Dedicated on-chip memory (OCM) interface
- Supports IBM CoreConnect™ bus architecture
- Debug and trace support
- Timer facilities

2.3 RocketIO 3.125 Gbps Transceivers

- Full-duplex serial transceiver (SERDES) capable of baud rates from 622 Mb/s to 3.125 Gb/s (please reference the Xilinx datasheet for speed grade limitations)
- 80 Gb/s duplex data rate (16 channels)

- Monolithic clock synthesis and clock recovery (CDR)
- Fibre Channel, Gigabit Ethernet, 10 Gb Attachment Unit Interface (XAUI), and
- Infiniband-compliant transceivers
- 8-, 16-, or 32-bit selectable internal FPGA interface
- 8B /10B encoder and decoder
- 50/75 on-chip selectable transmit and receive terminations
- Programmable comma detection
- Channel bonding support (two to sixteen channels)
- Rate matching via insertion/deletion characters
- Four levels of selectable pre-emphasis
- Five levels of output differential voltage
- Per-channel internal loopback modes
- 2.5V transceiver supply voltage

2.4 Virtex-II FPGA Fabric

Description of the Virtex-II Family fabric follows:

- SelectRAM memory hierarchy
  - Up to 10 Mb of True Dual-Port RAM in 18 Kb block SelectRAM resources
  - Up to 1.7 Mb of distributed SelectRAM resources
  - High-performance interfaces to external memory
- Arithmetic functions
  - Dedicated 18-bit x 18-bit multiplier blocks
  - Fast look-ahead carry logic chains
- Flexible logic resources
  - Up to 111,232 internal registers/latches with Clock Enable
  - Up to 111,232 look-up tables (LUTs) or cascadable variable (1 to 16 bits) shift registers
  - Wide multiplexers and wide-input function support
  - Horizontal cascade chain and Sum-of-Products support

- Internal 3-state busing

- High-performance clock management circuitry
  - Up to eight Digital Clock Manager (DCM) modules
    - Precise clock de-skew
    - Flexible frequency synthesis
    - High-resolution phase shifting
  - 16 global clock multiplexer buffers in all parts
- Active Interconnect technology
  - Fourth-generation segmented routing structure
  - Fast, predictable routing delay, independent of fanout
  - Deep sub-micron noise immunity benefits
- Select I/O-Ultra technology
  - Up to 852 user I/Os
  - Twenty two single-ended standards and five differential standards
  - Programmable LVTTL and LVCMOS sink/source current (2 mA to 24 mA) per I/O
  - Digitally Controlled Impedance (DCI) I/O: on-chip termination resistors for single-ended I/O standards
  - PCI support(1)
  - Differential signaling
    - 840 Mb/s Low-Voltage Differential Signaling I/O (LVDS) with current mode drivers
    - Bus LVDS I/O
    - HyperTransport™ (LDT) I/O with current driver buffers
    - Built-in DDR input and output registers
  - Proprietary high-performance SelectLink technology for communications between Xilinx devices
    - High-bandwidth data path
    - Double Data Rate (DDR) link
    - Web-based HDL generation methodology
- SRAM-based in-system configuration
  - Fast SelectMAP™ configuration

  - Triple Data Encryption Standard (DES) security option (bitstream encryption)
  - IEEE1532 support
  - Partial reconfiguration
  - Unlimited reprogrammability
  - Readback capability
- Supported by Xilinx Foundation™ and Alliance™ series development systems
  - Integrated VHDL and Verilog design flows
  - ChipScope™ Pro Integrated Logic Analyzer
- 0.13-µm, nine-layer copper process with 90 nm high-speed transistors
- 1.5V (VCCINT) core power supply, dedicated 2.5V VCCAUX auxiliary and VCCO power supplies
- IEEE 1149.1 compatible boundary-scan logic support
- Flip-Chip and Wire-Bond Ball Grid Array (BGA) packages in standard 1.00 mm pitch
- Each device 100% factory tested

3 Foundation ISE 6.1i

ISE Foundation is the industry's most complete programmable logic design environment. ISE Foundation includes the industry's most advanced timing driven implementation tools available for programmable logic design, along with design entry, synthesis and verification capabilities. With its ultra-fast runtimes, ProActive Timing Closure technologies, and seamless integration with the industry's most advanced verification products, ISE Foundation offers a great design environment for anyone looking for a complete programmable logic design solution.

3.1 Foundation Features

3.1.1 Design Entry

ISE greatly improves your "Time-to-Market", productivity, and design quality with robust design entry features. ISE provides support for today's most popular methods for design capture including HDL and schematic entry, integration of IP cores as well as robust support for reuse of your own IP. ISE even includes technology called IP Builder, which allows you to capture your own IP and reuse it in other designs.

ISE's Architecture Wizards allow easy access to device features like the Digital Clock Manager and Multi-Gigabit I/O technology. ISE also includes a tool called PACE (Pinout Area Constraint Editor), which includes a front-end pin assignment editor, a

design hierarchy browser, and an area constraint editor. By using PACE, designers are able to observe and describe information regarding the connectivity and resource requirements of a design, resource layout of a target FPGA, and the mapping of the design onto the FPGA via location/area.

This rich mixture of design entry capabilities provides the easiest to use design environment available today for your logic design.

3.1.2 Synthesis

Synthesis is one of the most essential steps in your design methodology. It takes your conceptual Hardware Description Language (HDL) design definition and generates the logical or physical representation for the targeted silicon device. A state of the art synthesis engine is required to produce highly optimized results with a fast compile and turnaround time. To meet this requirement, the synthesis engine needs to be tightly integrated with the physical implementation tool and have the ability to proactively meet the design timing requirements by driving the placement in the physical device. In addition, cross probing between the physical design report and the HDL design code will further enhance the turnaround time.

Xilinx ISE provides the seamless integration with the leading synthesis engines from Mentor Graphics, Synopsys, and Synplicity. You can use the synthesis engine of your choice. In addition, ISE includes Xilinx proprietary synthesis technology, XST. You have options to use multiple synthesis engines to obtain the best-optimized result of your programmable logic design.

3.1.3 Implementation and Configuration

Programmable logic design implementation assigns the logic created during design entry and synthesis into specific physical resources of the target device.

The term "place and route" has historically been used to describe the implementation process for FPGA devices and "fitting" has been used for CPLDs. Implementation is followed by device configuration, where a bitstream is generated from the physical place and route information and downloaded into the target programmable logic device.

To ensure designers get their product to market quickly, Xilinx ISE software provides several key technologies required for design implementation:

- Ultra-fast runtimes enable multiple "turns" per day
- ProActive™ Timing Closure drives high-performance results
- Timing-driven place and route combined with "push-button" ease
- Incremental Design

- Macro Builder

3.1.4 Board Level Integration

Xilinx understands the critical issues such as complex board layout, signal integrity, high-speed bus interface, high-performance I/O bandwidth, and electromagnetic interference for system level designers.

To ease the system level designers' challenge, ISE provides support to all Xilinx leading FPGA technologies:

- System IO
- XCITE
- Digital clock management for system timing
- EMI control management for electromagnetic interference

To really help you ensure your programmable logic design works in context of your entire system, Xilinx provides complete pin configurations, packaging information, tips on signal integration, and various simulation models for your board level verification including:

- IBIS models
- HSPICE models
- STAMP models

4 Virtex-II Pro Developer's Kit

V2PDK is the Virtex-II Pro Developer's Kit, and is included to provide an existing framework of hardware and software code to explore the capabilities of the Virtex-II Pro, as well as a basis to build new systems.

A wide variety of software and hardware tools are used to build a Virtex-II Pro™ design. V2PDK The design flow is a tool chain methodology that exists to simplify the entire design process by providing integration between the tools and automating tasks. The main focus of the design flow is integrating the programs with each other to accomplish the system design.

The system design process can be loosely divided into the following tasks:

- Builds the software application

- Simulates the hardware description
- Simulates the hardware with the software application
- Simulates the hardware into the FPGA using the software application in on-chip memory
- Runs timing simulation
- Configures the bitstream for the FPGA

Chapter 5

Introduction to the Reference Design

This chapter introduces the DN6000K10 Reference Design, including information on what the reference design does, how to build it from the source files, and how to modify it for another application.

1 Exploring the Reference Design

1.1 What is the Reference Design?

The reference design is a fully functional Virtex II Pro FPGA design capable of demonstrating most of the features available on the DN6000K10. Features exercised in the reference design include:

- Access to the DDR SDRAM Modules At 133Mhz
- Access to FLASH memory
- UART Communication
- FPGA Interconnect
- Interaction with the Configuration FPGA and MCU
- Use of Embedded PowerPC Processors
- Memory Mapped Access Between PPC And User Design
- Access to external LED's
- Communication via Rocket I/O Transceivers
- Instantiation of Daughter Card Test Headers

All source code for the reference design is included on the CD and may be used freely in customer development. Precompiled bit files for the most common stuffing options are also included and should be used to verify board functionality before beginning development. A build utility, described in the section Compiling The Reference Design, can be used to generate new bit files, or to generate bit files for less common configurations of the DN6000K10.

1.2 Using the Reference Design

For information on preparing the board for running the reference design, see Chapter 5: Programming / Configuring the Hardware. This section assumes that board has been set up with appropriate jumper settings and oscillators, code has been loaded for the Configuration FPGA and the MCU, and that the Reference Design has been loaded into at least FPGA A. **Note that when the board is shipped, all of these steps have already been completed- no modification to jumper settings, oscillators, Config FPGA code, or MCU code is required to use the Reference Design.**

The primary interface to the DN6000K10 Reference Design is through an RS232 Serial Port, connected to one of the four PPC RS232 headers, P3, P4, P6, and P7. For more information, see the section PPC RS232 Port Setup in Chapter 2: Getting Started, and the section Configuring HyperTerminal in Chapter 5: Programming / Configuring the Hardware. It is assumed at this point that a terminal emulator is connected to PPC Port1 (Header P3), running at 19200 bps.

Powering up the board will display the following text on the terminal:

```
****************************************
****************************************
** DN6000K10 ASIC DEVELOPMENT PLATFORM **
******* REFERENCE DESIGN SOFTWARE *******
****************************************
****************************************

FPGA_A:
Waiting for External Host Commands
Press Any Key To Enter Local User Menu
```

The various functions of the Reference Design may be controlled both from the MCU menu, described in the section Description of Main Menu Options in Chapter 5, or from the PowerPC menu. In this example we will be using the PowerPC menu to exercise the functions of the Reference Design. When presented with the above text, the Reference Design is waiting for commands to be sent from the MCU. Press any key to stop waiting for MCU commands and get the following menu:

```
********************
FPGA_A: MAIN MENU
********************

a) Run Full Test Suite
b) Test Registers
c) Test Flash
d) Test DDR
e) Test Interconnect
f) Write Memory Location
g) Read Memory Location
h) Display Memory in 8 DWORDS per Line Format
i) Fill Memory with specified DWORD pattern
j) Toggle Mem Owner: INTERNAL (User)
k) Interconnect Test Menu

q) Quit
```

Now tests can be run directly from the embedded PPC processor. The menu options are as follows:

a. Run Full Test Suite: Runs options b,c,d, and e
b. Test Registers: Runs read/write tests on local FPGA registers
c. Test Flash: Runs a full set of tests on the Flash (takes ~4 minutes)
d. Test DDR: Runs read/write tests on the DDR memories.
e. Test Interconnect: Runs an inter-FPGA test on the physical interconnect.
f. Write Memory Location: Allows writing to any PPC memory location
   DDR_BASE = 0x80000000
   FLASH_BASE = 0x90000000
   REGISTER BASE = 0x98000000
g. Read Memory Location: Allows reading any PPC memory location
h. Display Memory… : Starting from any PPC address, lists DWORDs
i. Fill Memory with specified DWORD pattern: Allows large chunks of memory to be filled with a known value.
j. Toggle Mem Owner: Sets the Memory Arbiter to User (PPC) or Host
k. Interconnect Test Menu: For interconnect debug (under construction)

Note that the full test suite takes about 5 minutes to run. To abort any test operation use the PPC Reset Button (S3) to reset the design.

1.3 Compiling The Reference Design

This section deals with the source code to the Reference Design, which can be found on the CD-ROM. All file references are with respect to the root directory of the Reference Design source code (/source/FPGA). Files that are specific to the DN6000K10 design are found in the DN6000K10 subdirectory, whereas general application code is found in the common subdirectory.

1.3.1 The Xilinx Embedded Development Kit (EDK)

The Reference Design uses the Xilinx EDK to instantiate an embedded PowerPC Processor. The EDK project can be found at 'DN6000K10/PPC/system.xmp' and can be opened and modified with the Xilinx Embedded Development Kit software.

1.3.2 Synplicity Synplify

The Dini Group uses Synplicity's Synplify software to for design synthesis. The Synplicity projects for each of the 9 FPGA's on the DN6000K10 can be found at 'DN6000K10/synthesis/*.prj'. These projects have been compiled using Synplify Pro version 7.3.

1.3.3 Xilinx ISE

A sample Project Navigator project is located at 'DN6000K10/implement/fpga.npl'. For information on using Xilinx ISE, see the section Foundation ISE 6.1i in Chapter 3.

1.3.4 The Build Utility: Make.bat

The Build Utility is found at 'DN6000K10/build/make.bat'. This batch file is used to set system parameters to the desired configuration (i.e. VP70 vs. VP100, DDR2 stuffed or not stuffed, etc.), and to invoke all of the above tools from the command line. Instructions for invoking the batch file can be found by viewing the batch file with a text editor. Additional information about using the batch file to build the reference design is found below. Taking the reference design through all of the various tools for several FPGA's can be very tedious and time consuming- this batch file can do it all in one command!

The command line utility "Make.bat" is an MS-DOS batch file compatible with Windows 2000 and later operating systems. Make.bat should be run from the command line, with command line parameters. It should not be double clicked from the windows environment. A command prompt shortcut is provided in the same directory as Make.bat, and can be double clicked to open a command prompt window with the proper working directory.

Four main steps are involved in building the reference design. First the PowerPC netlist must be built using the EDK. The first time this is done it must be done from the EDK GUI, not from the command line. Open the EDK project (in PPC/system.xmp), and select Tools->make netlist. Once this has been done once, the Make.bat script can be used to build the netlist with the command Make ppc_netlist. The second step is to synthesize the design with Synplicity's Synplify Pro. The third

step in to place and route, or "implement" the design with the Xilinx ISE tools. The fourth and final step is to compile the PowerPC code and embed it in the bitfile. This fourth step is referred to by Xilinx as "updating" the bitfile. Hence this fourth step will be referred to as the "update" step.

The build script creates a directory called "out" and places its output files there. After the script completes you will find 3 files for each FPGA that was built. Fpga_*.bit is the file to be downloaded to the FPGA. The fpga_*_ui.bit and the fpga_*.bmm files are used by the Xilinx EDK in the "update" process to embed the PowerPC source code into the bitfile, creating the final bitfile.

All of the steps mentioned above can be performed with the build script. The following command line options are supported:

All	Synthesizes, implements, and updates for all 9 fpga's. **Doesn't generate the PowerPC netlist**.
*	Replace * with A, B, C, D, E, F, G, H, or I. Synthesizes, implements, and updates for the specified FPGA
synthesize_*	Replace * with a,b,c,d,e,f,g,h,i, or all. Synthesizes the specified FPGA, or all FPGA's.
implement_*	Replace * with a,b,c,d,e,f,g,h,i, or all. Implements the specified FPGA, or all FPGA's.
update_*	Replace * with a,b,c,d,e,f,g,h,i, or all. Updates the specified FPGA, or all FPGA's.
Clean	Deletes all intermediate tool-generated files. Leaves out directory intact.
clean_all	Deletes all generated files accept those from the EDK
clean_ppc	Deletes all EDK netlist files
ppc_netlist	Rebuilds the EDK netlist. The netlist MUST previously have been build from the EDK user interface before it can be built from the command line.
make VP70	makes changes to synplicity and EDK project files, and UCF files to compile for VP70
make VP100	makes changes to synplicity and EDK project files, and UCF files to compile for VP100

make INCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to include DDR2
make EXCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to exclude DDR2
make DDR_32_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_64_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_128_MEG	makes changes to synplicity projects and EDK source code to set DDR size

The reference design must support any number of FPGA's in both VP70 and VP100 sizes. Compiler constants are used to include/exclude code, as well as to set appropriate parameters for the configuration being compiled for. Specifically, the user may want to include/exclude any memory device (DDR1, DDR2, FLASH), or may want to switch between the VP70 and VP100 part. There are four places where changes must be made to get the desired configuration:

I. Synplicity synthesis project file
II. UCF files in 'source/ucf'
III. Xilinx EDK project file
IV. Xilinx EDK processor source code files ('PPC/code/fpga_params/*.h')
V. Setting up the build utility: "make.bat"

Note that the build utility runs the xilinx tools from the command line, so there are no Xilinx Project Navigator files to edit. If you choose to use the Project Navigator GUI, be very careful to have all the appropriate settings (ie 2vp70 vs 2vp100) The following sections explain what to change and what options the user has to accomplish these changes (Most are automated, some are not). It is highly recommended that everything be recompiled after making any of these changes, including the PPC netlist, the synplicity project, the Xilinx project, and the EDK source code. If everything is not updated properly unpredictable behavior will result. If you aren't sure, delete all tool generated files and start fresh. For information on the usage of the build tool (make.bat), see the top of the 'make.bat' file.

I. SYNPLICITY SYNTHESIS PROJECT FILE

In the 'synthesis' folder there are nine project files, one for each FPGA. The line 'set_option -part XC2VP70' must be modified appropriately for the VP70 or VP100. This change, as well as changes to the parameters described below may be made through the build utility (described below), through the synplicity GUI, or by hand. At the bottom of each file is a list of defined compiler constants that dictate what code is included and what code is excluded. The recognized constants are as follows:

EXTERNAL_DEFINES	Tells 'fpga.v' not to define it's own set of constants, but to use the ones defined externally (by synplicity).
FPGA_X	Tells 'fpga.v' which FPGA is being compiled. 'X' must be replaced with A,B,C,D,E,F,G,H, or I. Used to define the FPGA's ID number and name for communication with the host.
VP70/VP100	Tells 'fpga.v' which fpga is being targeted. Used by the interconnect test to disable bus lines that are no connect in the VP70 part.
INTERCON_MASTER	This must be defined for one and only one FPGA in the system. It includes the control code for the interconnect test- all other FPGA's are passive in the interconnect test. Any FPGA can be the master, but only one!
INCLUDE_FLASH EXCLUDE_FLASH	Includes/Excludes the flash controller code.
INCLUDE_DDR1 EXCLUDE_DDR1	Includes/Excludes the ddr1 controller code.
INCLUDE_DDR2 EXCLUDE_DDR2	Includes/Excludes the ddr2 controller code.
DDR_32_MEG DDR_64_MEG DDR_128_MEG	Defines the size of the DDR's, for address mapping.

The above parameters may be modified by hand, or by using the build utility with the following options:

make VP70	makes changes to synplicity and EDK project files, and UCF files to compile for VP70

make VP100	makes changes to synplicity and EDK project files, and UCF files to compile for VP100
make INCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to include DDR2
make EXCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to exclude DDR2
make DDR_32_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_64_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_128_MEG	makes changes to synplicity projects and EDK source code to set DDR size

II. UCF FILES

In 'source/ucf' are 9 UCF files, one for each FPGA. The UCF files must be modified to exclude any unused memory device (DDR1, DDR2, or FLASH). If any DDR or FLASH chip is to be excluded, simply comment out all associated lines in the UCF file by putting a '#' in front of the line. If DDR2 is to be excluded (it should always be excluded for the VP70), then the build utility may be used as shown below. Use 'make VP70' or make 'VP100' to include/exclude bus interconnect lines that are appropriate to that device.

Please note that the bus numbering in the files under 'source/ucf' *does not* match the schematic. We have included a set of UCF files that do match the schematic, under the directory 'source/ucf_busnum_1toN'. The UCF files under that directory, however, will not work with the reference design. You may use them for your own design if you wish. The difference between the two versions is that the standard UCF files (source/ucf) have busses with numbering starting from 0, while the UCF files matching the schematic (source/ucf_busnum_1toN) have busses with numbering starting at 1.

make INCLUDE_DDR2:	makes changes to synplicity projects, EDK source code, and UCF files to include DDR2

make EXCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to exclude DDR2
make VP70	comments out bus interconnect lines that are 'No Connect' in the VP70
make VP100	uncomments bus interconnect lines that are 'No Connect' in the VP70, but exist in VP100

If excluding FLASH or DDR1, all changes must be made by hand (be sure to also make changes to the synplicity project file and the PPC source file 'PPC/code/fpga_params/*.h'

III. XILINX EDK PROJECT FILE

The Xilinx EDK Project file is found at 'PPC/system.xmp'. After making any changes to this file, be sure to select the 'clean all' option in the Xilinx EDK, so that all generated files will be remade with the new project settings. The only setting that should be changed in this file is the target device. This can be changed through the EDK GUI, using the build utility, or by hand. The device line looks like one of the following:

Device: xc2vp70

Device: xc2vp100

When changing between FPGA's the build utility can be used as follows:

make VP70	makes changes to synplicity and EDK project files and UCF files to compile for VP70
make VP100	makes changes to synplicity and EDK project files and UCF files to compile for VP100

IV. XILINX EDK PROCESSOR SOURCE CODE

The file 'PPC/code/fpga_params.h' defines the software parameters for the PowerPC part of the design. The folder 'PPC/code/fpga_parms' contains a parameter file for

each of the nine FPGA's. When compiling for FPGA_A, the file 'PPC/code/fpga_params/fpga_a.h' should be modified for the appropriate parameters, and then it's contents should be placed in 'PPC/code/fpga_params.h'. The build utility automatically copies the correct fpga parameters to fpga_params.h for each FPGA that it compiles. The parameters found in fpga_params.h (and each file in the fpga_params folder) are as follows:

FPGA_NAME	Defines text used in 'print' statements to identify the FPGA
INTERCON_MASTER	If INTERCON_MASTER was defined in the synplicity project file, then it should be defined here to include the associated menu options. See the synplicity project file section above for more information.
INCLUDE_FLASH	Includes menu options associated with the FLASH device
INCLUDE_DDR1 EXCLUDE_DDR1	Includes menu options associated with DDR memory
INCLUDE_DDR2 EXCLUDE_DDR2	Expands the DDR test range to twice the size.
DDR_32_MEG DDR_64_MEG	Defines DDR test range per DDR chip (define one or the other, or none if neither DDR is included)

These files may be editted by hand, or modified with the build utility as follows:

make INCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to include DDR2
make EXCLUDE_DDR2	makes changes to synplicity projects, EDK source code, and UCF files to exclude DDR2
make DDR_32_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_64_MEG	makes changes to synplicity projects and EDK source code to set DDR size
make DDR_128_MEG	makes changes to synplicity projects and EDK source code to set DDR size

V. Setting up the build utility: "make.bat"

The following tools must be installed on the system to use "make.bat":

- Xilinx ISE
- Xilinx EDK
- Synplicity Pro

In the section below %XILINX% should be replaced with your Xilinx install directory. By default this is "C:\Xilinx". %XILINX_EDK% should be replaced with your Xilinx EDK install directory. This is commonly "C:\Xilinx\EDK". %SYNPLICITY% should be replaced with your synplicity install directory. This is usually of the form "C:\Program Files\Synplicity\synplify_XX" where XX is the version number, like synplify_76 for synplify version 7.6.

The following directories must be in your "Path" environment variable:

- %XILINX%\bin\nt
- %XILINX_EDK%\gnu\powerpc-eabi\nt\bin;
- %XILINX_EDK%\xygwin\bin;
- %SYNPLICITY%\bin

At the bottom of each .prj file in the "synthesis" directory, is a line with the format:

- set_option -include_path "..."

Add the path "%SYNPLICITY%/lib/xilinx/" to this line if it is not already there.

2 Getting More Information

2.1 Printed Documentation

The printed documentation, as mentioned previously, takes the form of a Virtex-II Pro datasheet and a DN6000K10 User Guide.

2.2 Electronic Documentation

Multiple documents and datasheets have been included on the CD.

2.3 Online Documentation

There is a public access site that can be found on the Dini Group web site at http://www.dinigroup.com/.

Chapter 6

Programming/Configuring the Hardware

This chapter details the programming and configuration instructions for the DN6000K10.

1 Programming the Configuration FPGA

Note: The Configuration FPGA/PROM only needs to be programmed when an update is required.

Code updates will be posted on the Dini Group website. The user is required to purchase the Xilinx Development Tools if in-house development is required. The tools are available from Xilinx, (http://www.xilinx.com/).

The Configuration FPGA (U13) is programmed using an in-system programmable configuration PROM (U12). The JTAG chain from the PROM is in a serial daisy chain with the Configuration FPGA, allowing simultaneous JTAG programming option of both devices. The Configuration FPGA is set to Master Serial Mode using dipswitch (S4). At power-up, the Configuration FPGA provides a configuration clock (CFPGA_CCLK) that drives the PROM. A short access time after CEn (CFPGA_DONE) and OE (CFPGA_INITn) are enabled, data is available on the PROM data (CFPGA_D0) pin that is connected to the Configuration FPGA. The programming header (J7) as shown in Figure 18, is used to download the files to the Configuration PROM/FPGA via a Xilinx Parallel IV cable.

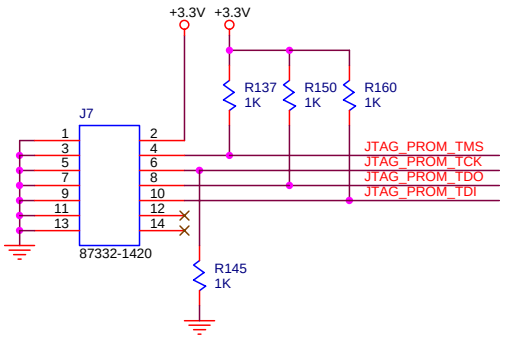


Figure 4 – Configuration PROM/FPGA Programming Header

This section lists detailed instructions for programming the Configuration FPGA PROM using the Xilinx ISE 6.1i tools.

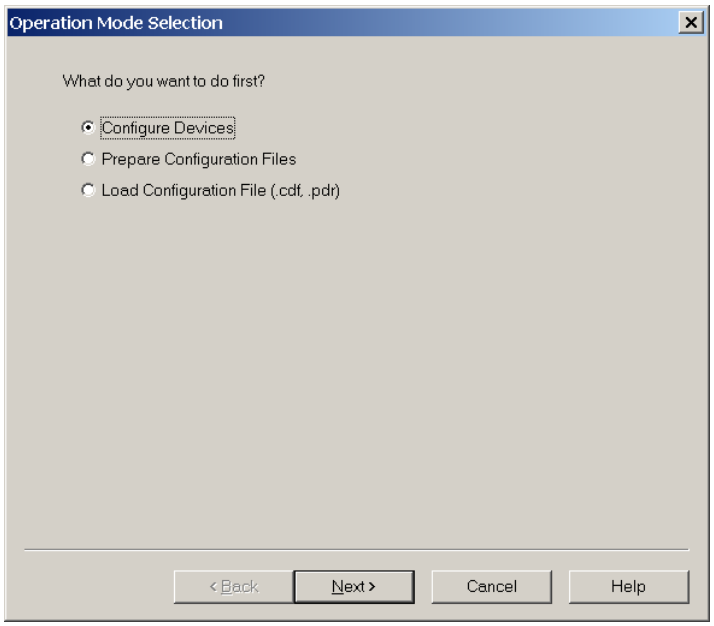
Note: This user guide will not be updated for every revision of the Xilinx tools, so please be aware of minor differences.

1. The DN6000K10 must be powered with the Xilinx JTAG cable connected to header J7 and the other end to a parallel port on the PC.

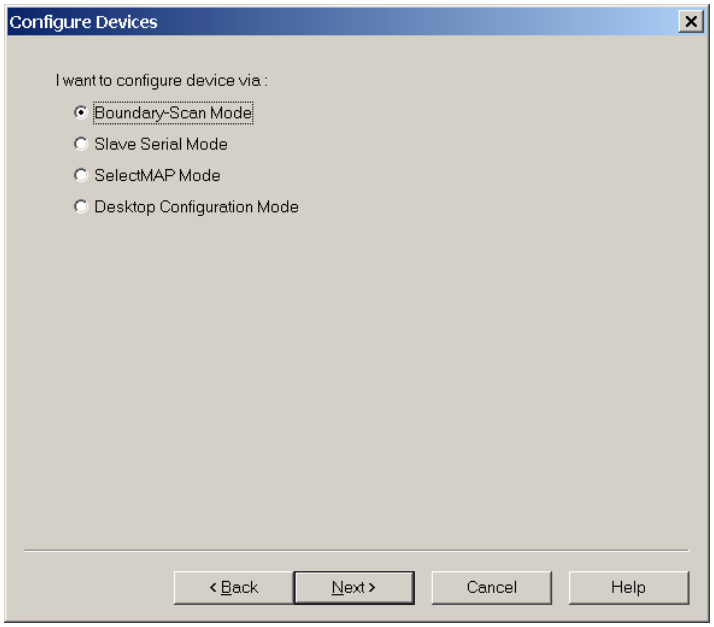
2. Download the latest programming file for the Configuration FPGA from the Dini Group website (filename "Prom.MCS") http://www.dinigroup.com/.

3. Run iMPACT - From the Windows **START** menu, choose **PROGRAMS → Xilinx ISE 6 → Accessories → iMPACT.**

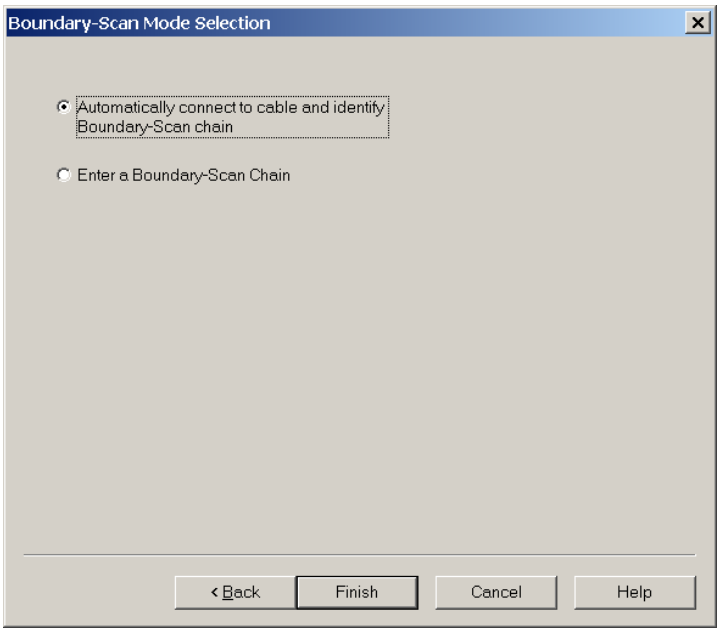
4. Select the **Configure Devices** option and proceed by clicking the **NEXT** button**.**



5. Select the **Boundary-Scan Mode** option and proceed by clicking the **NEXT** button**.**



6. Select the **Automatically connect to cable and identify Boundary-Scan chain** option and proceed by clicking the **NEXT** button**.**

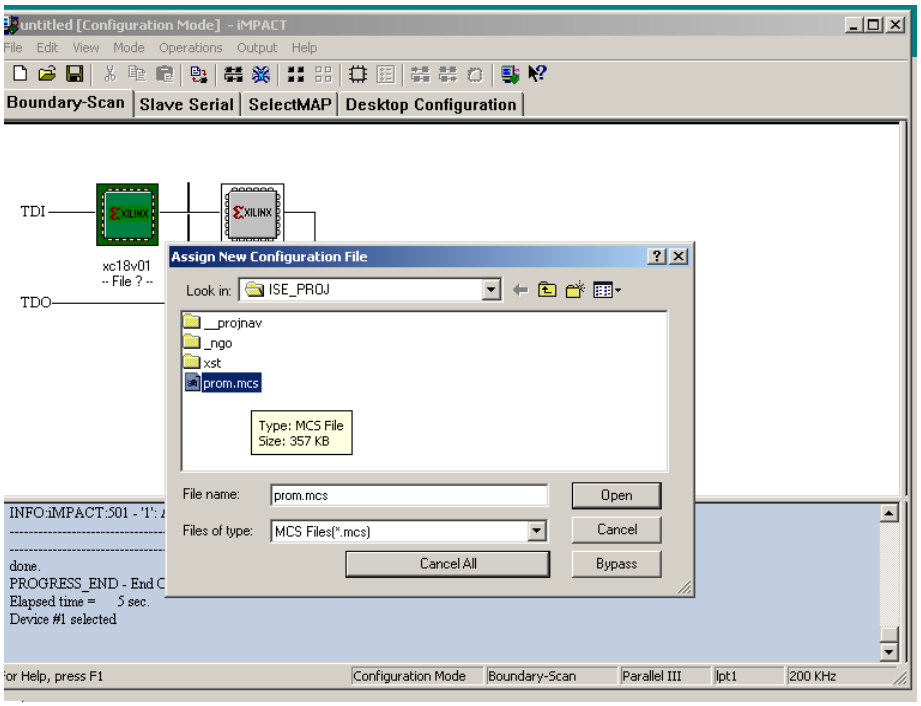


7. If the process was successful the following window will appear:

Boundary-Scan Chain Contents Summary
There was one device detected in the boundary-scan chain.
iMPACT will now direct you to associate a programming or
BSDL file with this device...
OK

8. Click **OK** button.

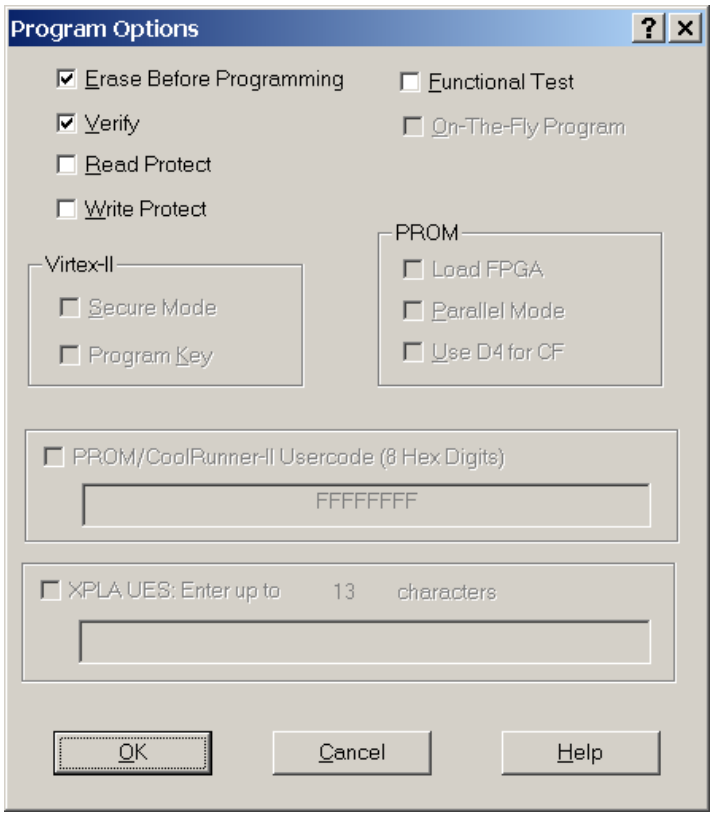
9. Enter the location of the PROM.MCS file in the window prompting the file name and click **OK**. Select Bypass for the second device in the chain (XC2S150). The following window would be displayed:



Note: Two devices should be detected, XC18V01 and XC2S150.

10. Select the XC18V01 right click and select **Program** option. The XC2S150 is not programmed.

11. Select the **Erase before programming** and the **Erase** option before clicking the **OK** button.



12. The Configuration FPGA is now programmed. You must power cycle the board before the Configuration FPGA will be configured with the new PROM data.

2 MCU Details / Programming the MCU

Switch 4 on S2 tells MCU how to boot

- If the 4th switch position is ON then the MCU boot sequence will behave in the following manner:
  - (1) If the USB cable is plugged in when the DN6000k10 is powered-on/reset the MCU boots from the EEPROM (U8) and waits for USBController applicatin to send commands. In this case, the MCU FLASH firmware stored in U6 can be updated. In this state the MCU has limited USB functionality and cannot configure the FPGAs via USB/SmartMedia or perform many of the other USB GUI functions.
  - (2) If the USB cable is NOT plugged in when the DN6000k10 is powered-on/reset the MCU first boots from the EEPROM

(U8) and then automatically boots from the MCU FLASH (U6). In this case, the MCU FLASH can NOT be updated.

- If the 4th switch position is OFF then the MCU will always boot from the MCU FLASH (U6) regardless of whether the USB cable is plugged in or not. When the MCU has booted from the FLASH it has full USB and FPGA configuration functionality. This is the default factory setup as of 1/1/05. Please note you can NOT update the MCU FLASH in the switch position.

3 Configuring HyperTerminal

A terminal emulator is required to monitor MCU transactions and to interact with the embedded PowerPC processors in the Reference Design. The Dini Group suggests using the Windows-based program - HyperTerminal (Hypertrm.exe). The configuration files for HyperTerminal "mcu_rs232.ht" and "ppc_rs232.ht" are supplied on the CD-ROM or can be downloaded from the Dini Group website.

The RS232 ports are configured with the following parameters:

- Bits per second: 19200
- Data bits: 8
- Parity: None
- Stop Bits: 1
- Flow control: None
- Terminal Emulation: VT100

Two cables converting the 5 x 2 header to a DB9 are shipped with the DN6000K10. The 5 x 2 headers connect to the MCU RS232 header P2, and any of the four PPC RS232 headers P3, P4, P6, and P7. These headers are not keyed - ensure correct pin orientation as noted below.

Note: MCU RS232 Header P2 is not keyed. Ensure correct pin orientation. Pin 1 is indicated with a letter 1 on the board silkscreen, as well as a dot. Pin 1 on the 5 X 2 cable header is indicated with a triangular shape printed on the connector, and by a colored wire on the cable.

Two female-to-female RS232 cables are provided with the DN6000K10. These cables will attach directly to the RS232 ports of a PC. The Dini Group suggests Jameco as a possible supplier, (http://www.jameco.com). The part number is 132345. Male-to-female extension cables are part number 25700.

4 Configuring the FPGA using SelectMAP

The simplest mode of configuration for the DN6000K10 Virtex-II PRO FPGA involves the SelectMAP configuration method using a SmartMedia card. The DN6000K10 ships with two 32 MB SmartMedia cards. One of these SmartMedia cards contains reference design bit files produced for SelectMAP configuration, and a file named "main.txt" that sets the configuration options (see "Creating Configuration File main.txt"). The other SmartMedia card is empty and available for user applications. To configure the FPGA's with the reference design, please skip to "Starting SelectMAP Configuration".

Status messages are reported by the MCU via the RS232 serial port during FPGA configuration. It is **NOT** necessary to have the serial port connection in order to configure the FPGA's in SelectMAP mode. However, if an error occurs during the configuration, the user would be able to identify possible problems by viewing the configuration status messages. See Configuring HyperTerminal on how to setup the serial port.

4.1 Bit File Generation for SelectMAP Configuration

Configuring the DN6000K10 Virtex-II PRO FPGA requires the generation of bit files by the Xilinx ISE tools.

NOTE: This user guide will not be updated for every revision of the Xilinx tools, so please be aware of minor differences. The Xilinx ISE 6.1i revision is used here.

First, a project must be created. Open the Xilinx ISE **Project Navigator** software package. Go to the File menu and select **New Project**. A "New Project" dialog box will pop up shown in Figure 7.

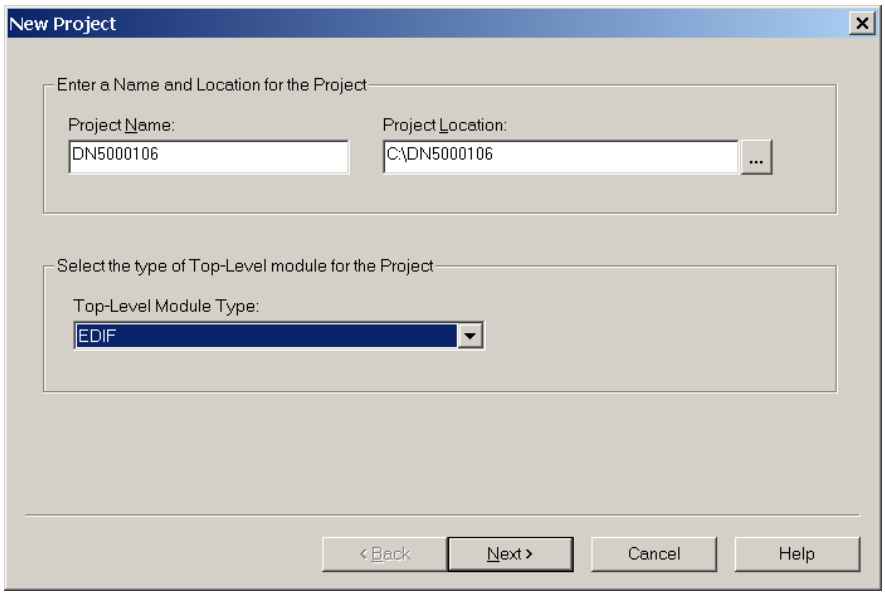


Figure 5 - New Project Screen Shot

Select the input files for the project, refer to Figure 6.

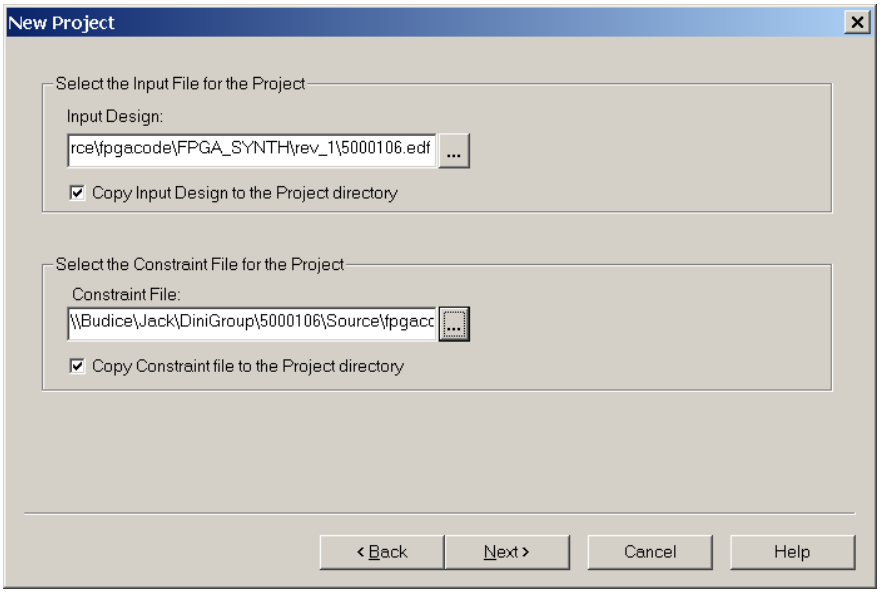


Figure 6 - Input File

Select the device and the design flow for the project. The user must specify a project name and location. The correct property values must be selected, refer to Figure 7:

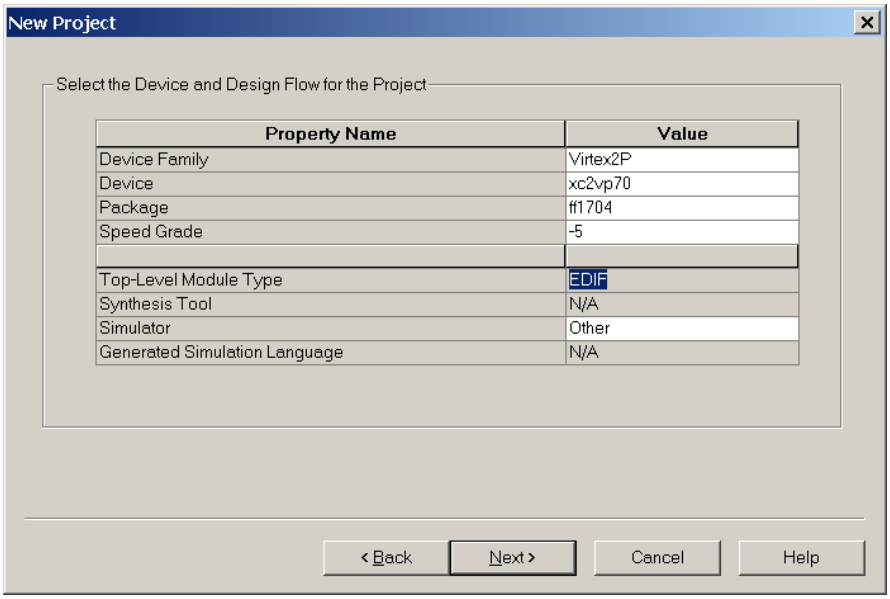


Figure 7: New Project Dialog Box

The Project Navigator will create a new project with the required files.

The DINI Group prefers to use Synplicity's Synplify for synthesis (which is recommended for the user also). Consequently, edif files are used in the design flow described here.

Selecting the edif file in the "Module View" window, the user's Project Navigator box should resemble Figure 8.

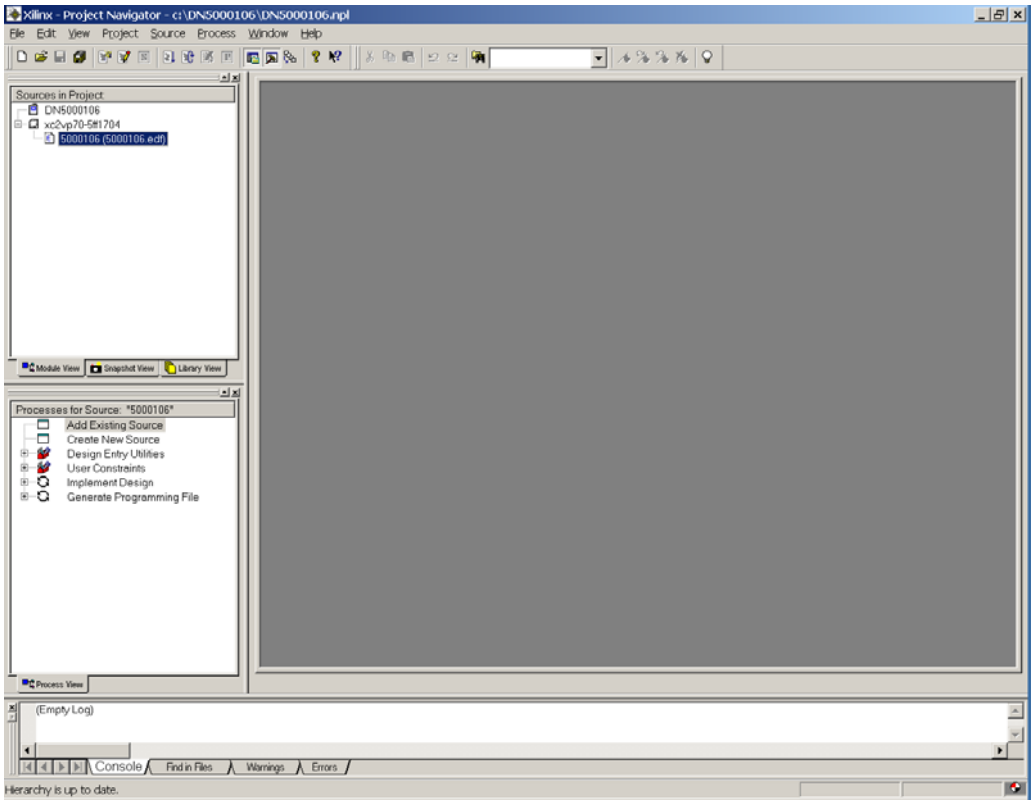
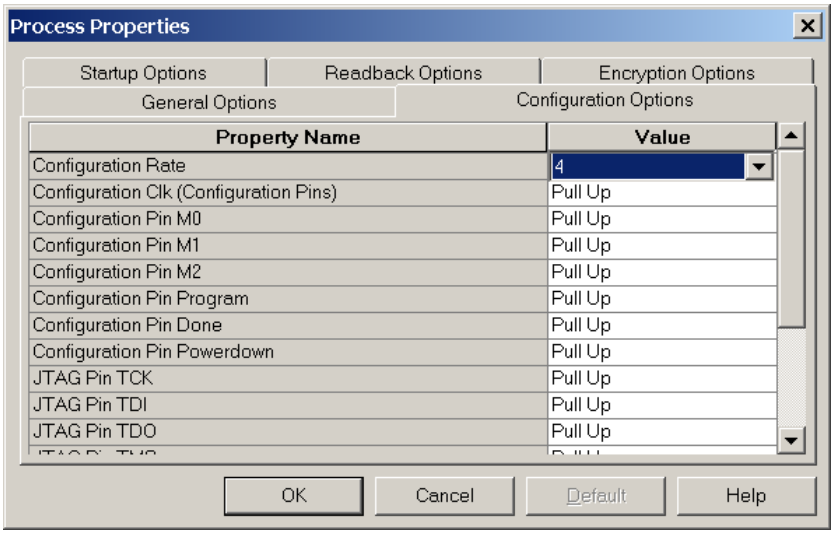


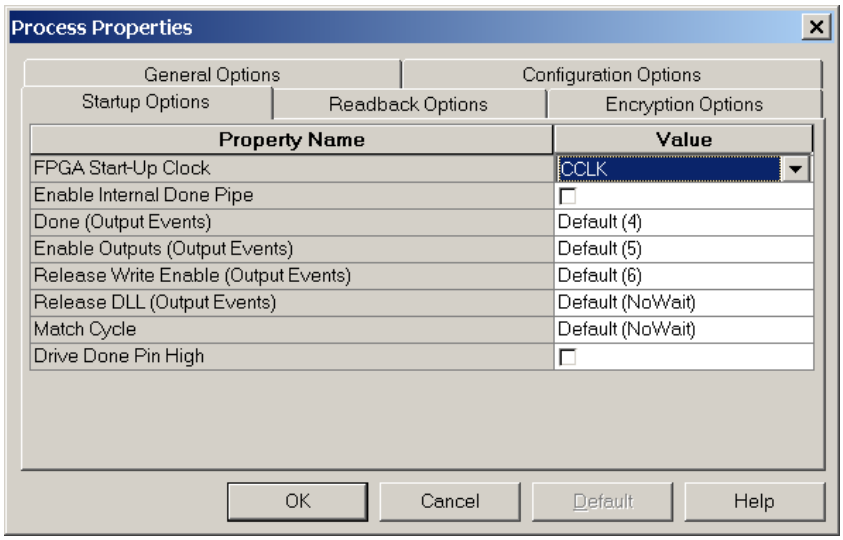
Figure 8: Project Navigator

In the "Process for Source" window, a process is signified by the icon . In the "Process for Source" window, the user must right-click on the "Generate Programming File" process and select properties. The default settings are correct (The user should verify a couple important options, right-click and selecting properties options).

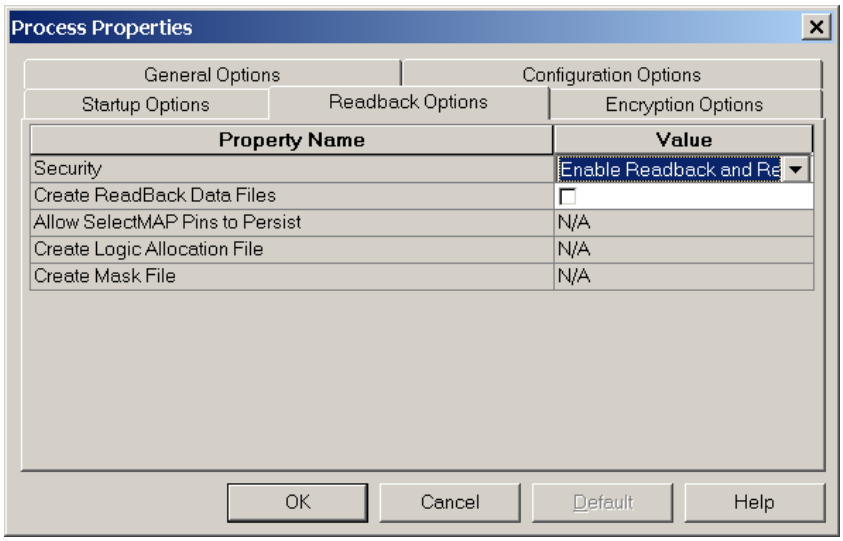
- Configuration Options Tab: Configuration Pin Powerdown = **Pull Up**



- Startup Options Tab: FPGA Start-up Clock = **CCLK**



- Readback Options Tab: Security = **Enable Readback and Reconfiguration**



The user can now generate the bit file. In the "Process for Source" window, the user must right-click on the "Generate Programming File" process and select Run. The bit file will be generated and may be found in the project directory.

4.2 Creating Configuration File "main.txt"

To control which bit file on the Smart Media card is used to configure which FPGA in SelectMAP mode a file named "main.txt" must be created and copied to the root directory of the Smart Media card. The configuration process cannot be performed without this file. Below is a description of the options that can be set in the file, a description of the format this file needs to follow, and an example of a main.txt file.

4.2.1 Verbose Level

During the configuration process, there are three different verbose levels that can be selected for the serial port messages:

- Level 0:
  - Fatal error messages
  - Bit file errors (e.g., bit file was created for the wrong part, bit file was created with wrong version of Xilinx tools, or bitgen options are set incorrectly)
  - Initializing message will appear before configuration
  - A single message will appear once the FPGA is configured
- Level 1:
  - All messages that Level 0 displays
  - Displays configuration type (should be SelectMAP)

  - Displays current FPGA being configured if the configuration type is set to SelectMAP
  - Displays a message at the completion of configuration for each FPGA configured.
- Level 2:
  - All messages that Level 1 displays
  - Options that are found in "main.txt"
  - Bit file names for each FPGA as entered in main.txt
  - Maker ID, device ID, and size of Smart Media card
  - All files found on Smart Media card
  - If sanity check is chosen, the bit file attributes will be displayed (part, package, date, and time of the bit file)
  - During configuration, a "." will be printed out after each block (16 KB) has successfully been transferred from the Smart Media to the current FPGA

4.2.2 Sanity Check

The Sanity Check, if enabled, verifies that the bit file was created for the right part, the right version of Xilinx was used, and the bitgen options were set correctly. If any of the settings found in the bit file are not compatible with the FPGA, a message will appear from the serial port, and the user will be asked whether or not they want to continue with the bit file. Please see the section Bit File Generation for SelectMAP Configuration for details on which bitgen options need to be changed from the default settings..

4.2.3 Format of "main.txt"

The format of the main.txt file is as follows:

1. The first nonempty/uncommented line in main.txt should be:

   Verbose level: X

   where "X" can be 0, 1 or 2. If this line is missing or X is an invalid level, then the default verbose level will be 2.

2. The second nonempty/uncommented line in main.txt tells whether or not to perform a sanity check on the bit files before configuring an FPGA:

   Sanity check: y

   where "y" stands for yes, "n" for no. If the line is missing or the character after the ":" is not "y" or "n" then the sanity check will be enabled.

3. For each FPGA that the user wants to configure, there should be exactly one entry in the main.txt file with the following format:

   FPGA F: example.bit

   In the above format, the "F" following FPGA is to signal that this entry is for FPGA F, and FPGA F would then be configured with the bit file example.bit. The DN6000K10 only has one FPGA, which is FPGA F. There can be any number of spaces between the ":" and the configuration file name, but they need to be on the same line.

4. Comments are allowed with the following rules:

   - All comments must start at the beginning of the line.
   - All comments must begin with //
   - If a comment spans multiple lines, then each line should start with //

   Commented lines will be ignored during configuration, and are only for the user's purpose.

5. The file main.txt is NOT case sensitive.

6. Example of "main.txt":

```
//start of file "main.txt"
Verbose level: 2
Sanity check: y
FPGA F: fpgaF.bit
//the line above configures FPGA F with the bit file "fpgaF.bit"
//end of main.txt
```

Given the above example file: Verbose level is set to 2, a sanity check on the bit files will be performed, and FPGA F will be configured with file fpgaF.bit.

NOTE: All configuration file names have a maximum length of eight (8) characters, with an additional three for the extension. Do not name your configuration bit files with long file names. In addition, all file names should be located in the root directory of the Smart Media card—no subdirectories or folders are allowed. Since the "main.txt" file controls which bit file is used to configure the FPGA, the Smart Media card can contain other bit files.

4.3 Starting SelectMAP Configuration

If using the reference design SmartMedia card that came with the DN6000K10 then no files need to be copied to the card. Otherwise, copy your bit files and "main.txt" to the root directory of the SmartMedia card using the FlashPath floppy adapter or some other means. Make sure the dipswitch (S2) is set for SelectMAP as shown in Table 2.

Table 2: S2 Dipswitch Configuration Settings

Signal Name	Pins	Status
FPGA_MSEL0	Pins 1 & 8	Closed
FPGA_MSEL1	Pins 2 & 7	Open
FPGA_MSEL2	Pins 3 & 6	Open
DIP_SW3	Pins 4 & 5	X

Set up the serial port connection as described above in Configuring HyperTerminal. Next, place the SmartMedia card in the SmartMedia socket on the DN6000K10 and turn on the power (NOTE: the card can only go in one way). The SmartMedia card is hot swappable and can be taken out or put into the socket even when the power is on. Once the power has been turned on, the configuration process will begin as long as there is a valid SmartMedia card inserted properly in the socket.

A SmartMedia card is determined to be invalid if either the format of the card does not follow the SSFDC specifications, or if it does not contain a file named main.txt in the root directory. If the configuration was successful, a message stating so will appear and the Main Menu will come up. Otherwise, an error message will appear. The LED's on DS1 and DS2 give feedback during and after the configuration process (see Table 23 for GPIO LED's for further details).

After the FPGA has been configured, the following Main Menu will appear via the serial port, refer to Figure 9.

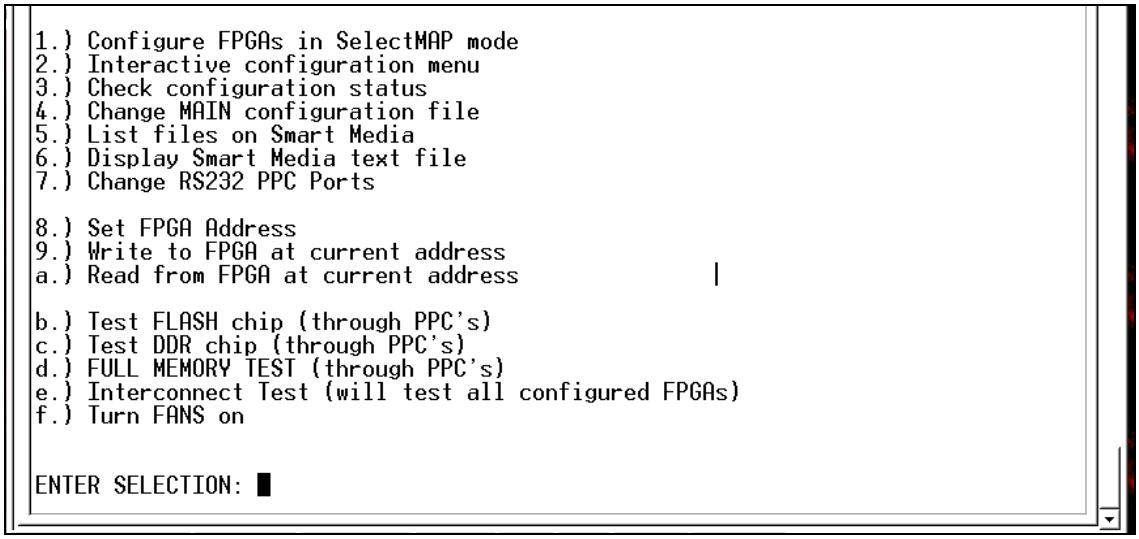

1.) Configure FPGAs in SelectMAP mode
2.) Interactive configuration menu
3.) Check configuration status
4.) Change MAIN configuration file
5.) List files on Smart Media
6.) Display Smart Media text file
7.) Change RS232 PPC Ports
8.) Set FPGA Address
9.) Write to FPGA at current address
a.) Read from FPGA at current address
b.) Test FLASH chip (through PPC's)
c.) Test DDR chip (through PPC's)
d.) FULL MEMORY TEST (through PPC's)
e.) Interconnect Test (will test all configured FPGAs)
f.) Turn FANS on
ENTER SELECTION:

Figure 9 - Main Menu

The HyperTerminal interface gives the user an easy method for handling and monitoring the DN6000K10 FPGA configuration.

4.3.1 Description of Main Menu Options

Table 3 describes the Main Menu options found on the MCU HyperTerminal interface.

Table 3: HyperTerminal Main Menu Options

Option	Function	Description
1	Configure FPGA's Using "main.txt"	The FPGA will configure in SelectMAP mode.
2	Interactive FPGA configuration menu	This option takes you to a menu titled "Interactive Configuration Menu" and allows the FPGA's to be configured through a set of menu options instead of using the main.txt file. The menu options are described below.
3	Check Configuration Status	This option checks the status of the DONE pin and prints out whether or not the FPGA's have been configured along with the file name that was used for configuration.
4	Change MAIN configuration file	By default, the processor uses the file main.txt to get the name of the bit file to be used for configuration as well as options for the configuration process. However, a user can put several files that follow the format for main.txt on the SmartMedia card that contain different options for the configuration process. By selecting the main menu option 4, the user can select a file from a list of files that can be used in place of main.txt. If the power is turned off or the reset button (S1) is pressed, the configuration file is changed back to the default, main.txt.

Option	Function	Description
5	List files on SmartMedia	This option prints out a list of all the files found on the SmartMedia card.
6	Display Contents of a TXT File	This option allows the user to list the contents of any text file on the Smart Media card.
7	Change RS232 PPC Ports	This options allows the user to select what FPGA PPCs should be connected to which PPC PORTS (P3, P4, P6, & P7). This option will also print out the current port settings allowing you to quit without changing them.
The next 7 options are only available if the FPGAs are configured with The Dini Group reference design. Please see Appendix A for FPGA Address Maps.		
8	Set FPGA Address	Set the fpga address for the next read/write to the fpga.
9	Write to FPGA at current address	Performs a DWORD write to the current fpga address. You will see the current address at the top of the Main Menu and also the write data after selection this option.
a	Read from FPGA at current address	Performs a DWORD read at the current FPGA address. You will see the current address and readback data at the top of the Main Menu.
b	Test FLASH Chip (through PPC's)	Allows the user to select which FPGA/FLASH to test. The test is actually run by the PPC's and all detailed test messages will appear on PPC PORT 1 (P3)
c	Test DDR Chip (through PPC's)	Allows the user to select which FPGA/DDR(s) to test. The test is actually run by the PPC's and all detailed test messages will appear on PPC PORT 1 (P3)

d	FULL MEMORY TEST (through PPC's)	Runs the following tests on all configured FPGAs: DDR, FLASH, Internal Registers, and Interconnect. The FPGA tests are performed in parallel and the user needs to hook up the PPC Ports to see detailed test messages. Only 4 FPGAs can output test messages via the PPC Ports so they will need to be setup before hand.
e	Interconnect Test	Runs the interconnect test on all configured FPGAs
f	Turn fans on/off	Either turns the fans on/off depending on current setting

Selecting "Option 2" results in the following menu to be displayed, refer to Figure 10.

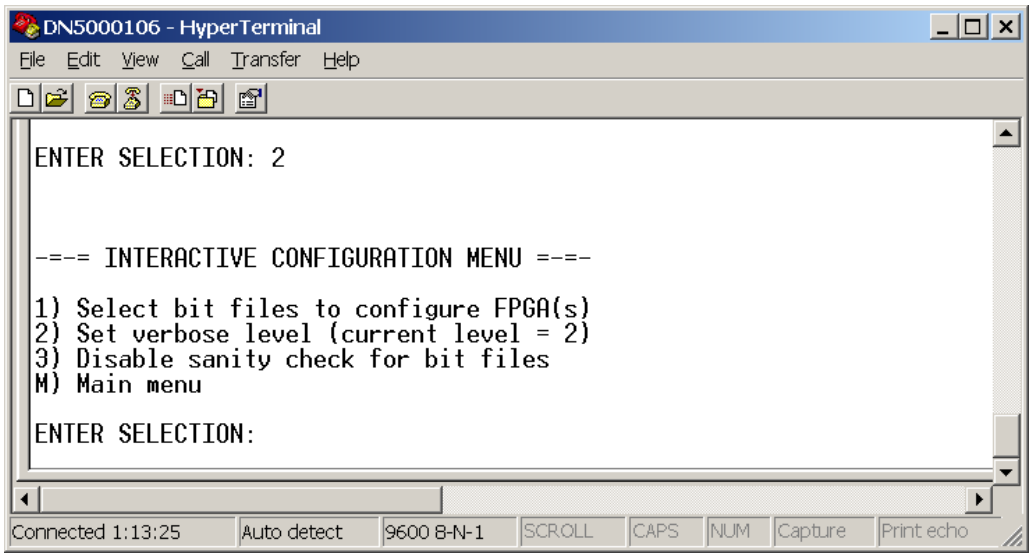


Figure 10 - Interactive Configuration Option Menu

Table 4 describes the Interactive Configuration Menu options:

Table 4: HyperTerminal Interactive Configuration Menu Options

Option	Function	Description
1	Select a bit file to configure FPGA(s)	The user is able to select a bit file from a list of bit files found on the SmartMedia card for configuring the FPGA.
2	Set verbose level (current level = 2)	The user can change the verbose level from the current setting. NOTE: If the user goes back to the main menu and configures the FPGA(s) using main.txt, the verbose level will be set to whatever setting is specified in main.txt.
3	Disable/Enable sanity check for bit files	The user can disable or enable the sanity check, depending on what the current setting is. NOTE: If the user goes back to the main menu and configures the FPGA(s) using main.txt, the sanity check will be set to whatever setting is specified in main.txt.
M	Main menu	Returns the user to the Main Menu.

4.4 Bitstream Encryption

Virtex-II Pro devices have an on-chip decryptor using one or two sets of three keys for triple-key Data Encryption Standard (DES) operation. Xilinx software tools offer an optional encryption of the configuration data (bitstream) with a triple- key DES determined by the designer. The keys are stored in the FPGA by JTAG instruction and retained by a battery connected to the VBATT pin, when the device is not powered. Virtex-II Pro devices can be configured with the corresponding encrypted bitstream, using any of the configuration modes described previously. A detailed description of how to use bitstream encryption is provided in the *Virtex-II Pro Platform FPGA User Guide*.

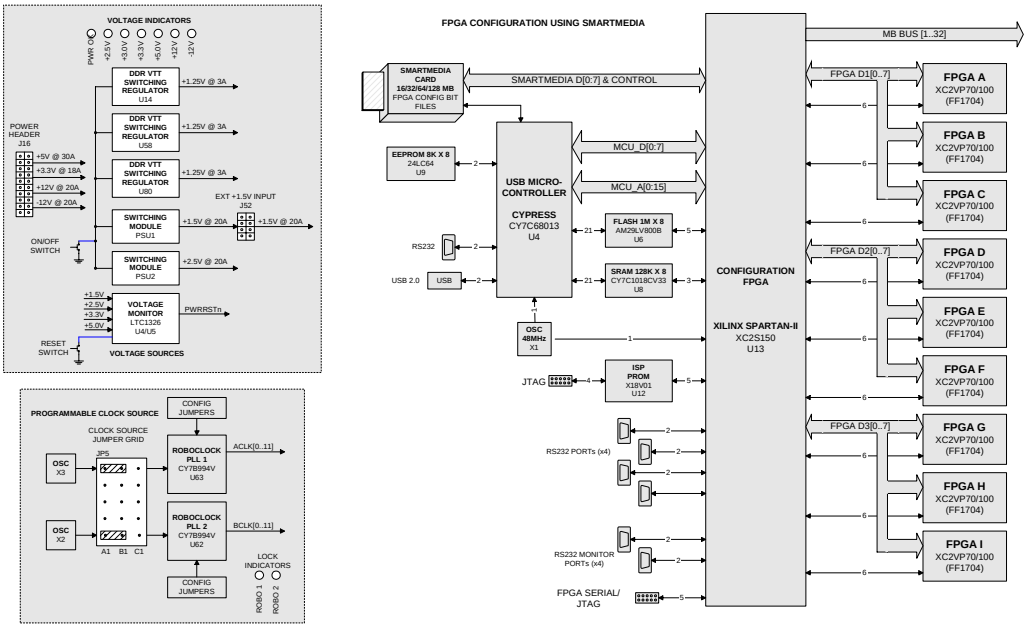
Chapter 7

Board Hardware

1 Introduction to the Board

DN6000K10 Logic Emulation board provides for a comprehensive collection of peripherals to use in creating a system around the Virtex-II Pro FPGA. Figure 11 is a block diagram of the DN6000K10 Logic Emulation board. FIXME below diagram

DN6000K10 BLOCK DIAGRAM



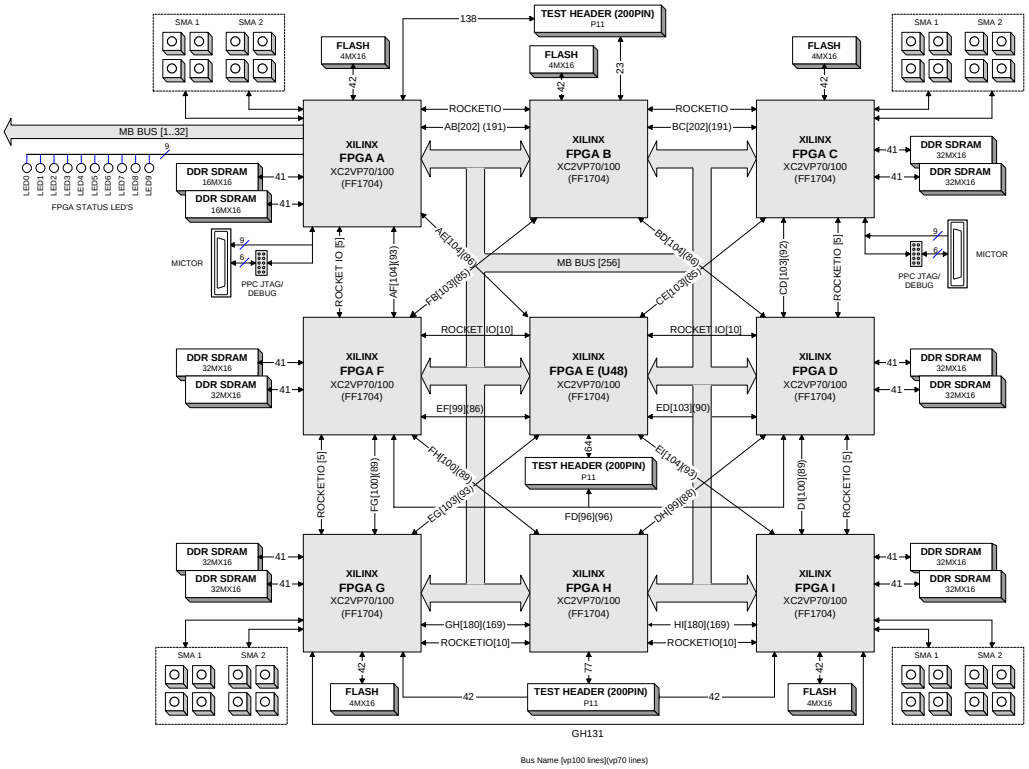


Figure 11 - DN6000K10 Block Diagram

1.1 DN6000K10 Functionality

The components and interfaces featured on the DN6000K10 includes:

- 2VP70/100 Virtex-II Pro FPGA Options (x 9)
- Flexible and Configurable Clocking Scheme (RoboClockII)
- SmartMedia FPGA Configuration
- USB2.0 Interface
- DDR SDRAM, 16M x 16 (2 per FPGA)
- FLASH, 4M x 16 (x 3)
- Two Multi-Gigabit Transceiver (MGT) channels (SMB) / FPGA (A, C, G, I)
- One User Clock SMA Interface (differential SMB)
- 200 Pin Test Header (x 3)

- CPU Debug and Trace Interfaces, in Berg and Mictor connectors
- ATX Power Supply Connection

NOTE: RocketIO interface speed is directly affected by the speed grade of the FPGA. Please refer to the Xilinx datasheet.

2 Virtex-II Pro FPGA

The Virtex-II Pro FPGA's are situated on the topside of the board. For a detailed description of the capabilities of the Virtex-II Pro FPGA's, refer to the datasheet on the Xilinx website.

2.1 FPGA (2VP70) Facts

The Virtex-II Pro Platform FPGA'a on board the DN6000K10 is in the FF1704 package. The capabilities of the 2VP70 (base model) include:

- 2 PowerPC™ 405 processor
- 16 or 20 Multi-Gigabit Transceivers (MGTs)
- 996 SelectI/O
- 8 Digital Clock Managers (DCMs)
- ~33000 logic slices
- ~5900 Kbits of BlockRAM (BRAM)
- 328 18 x 18-bit multiplier blocks

The FF1704 package on the DN6000K10 is a 1.0mm (42.5 x 42.5mm) fully populated (with four corner balls removed) flip chip BGA.

The PowerPC™ 405 is capable of operation at 300+ MHz, and is capable of 420+ Dhrystone MIPs (dependent on the speed grade of the part). Each of the MGTs are capable of 3.125 Gigabits per second in both directions, for an aggregate bandwidth of 50 Gigabits per second from the MGTs (25 Gbps transmit and 25 Gbps receive). The SelectIO are capable of supporting multiple high-speed I/O standards, from LVDS to SSTL2 to PCI. The DCMs are capable of 24 MHz to 420 MHz operation and provide for clock deskew, frequency synthesis, and fine phase shifting.

3 FPGA Configuration

The Dini Group developed the SmartMedia Configuration Environment to address the need for a space-efficient, pre-engineered, high-density configuration solution for systems with single or multiple FPGA's. The technology is a groundbreaking in-system programmable configuration solution that provides substantial savings in development effort and cost per bit over traditional PROM and embedded solutions for high-capacity FPGA systems.

Virtex-II Pro devices are configured by loading application-specific configuration data into internal memory. Configuration is carried out using a subset of the device pins, some of which are dedicated, while others can be reused as general-purpose inputs and outputs after configuration is complete. SmartMedia is the primary means of configuring the FPGA's on the DN6000K10 board. Configuration of FPGA's is accomplished using either Serial/SelectMAP or the JTAG interface. The remainder of this section describes the functional blocks that entail the FPGA configuration environment.

3.1 Micro Controller Unit (MCU)

The Cypress CY7C68013 (U7) micro controller is used to control the configuration process. The MCU contains an enhanced 8051 core, USB 2.0 transceiver and a Serial Interface Engine (SIE). The CY7C68013 provides the following features: 256 bytes of register RAM, three flexible Timers, 2 USARTs, and an integrated I^2C compatible controller.

The MCU interfaces to the Configuration FPGA (U13) via an 8-bit bus and the SmartMedia interfaces to the Configuration FPGA via an 8-bit bus. The FPGA's (x9) on the board interfaces to the Configuration FPGA via the JTAG interface and an 8-bit bus, used during Serial and SelectMap programming of the FPGA's. The amount of internal SRAM is not large enough to hold the FAT needed for SmartMedia, so an external 128Kb x 8 SRAM (U8) was added. In addition a 1Mb x 8 FLASH (U6) was added to store the downloaded program code. An external EEPROM (U9) configures the MCU during power-up.

The micro controller has the following responsibilities:

- Reading the SmartMedia card via the Configuration FPGA
- Communicate to the system via the USB Interface
- Configuring the Virtex-II Pro FPGA's (9)
- Executing DN6000K10 self tests
- Drive status LED's

3.1.1 MCU EEPROM Interface

During the power-up sequence, internal logic checks the I²C-compatible port for the connection of an EEPROM (U9) whose first byte is either 0xC0 or 0xC2. If found the MCU uses the VID/PID/DID values in the EEPROM in place of the internally stored values of it boot-loads the EEPROM contents into internal RAM (0xC2). The EEPROM interface is shown in Figure 12.

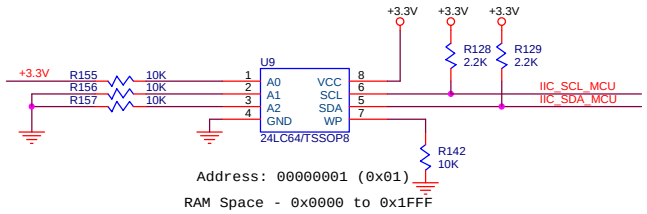


Figure 12 - MCU EEPROM Interface

3.1.2 MCU SRAM External

Memory expansion for the MCU is provided as 128k x 8 SRAM (U8). Writing to the device is accomplished by taking Chip Enable (SRAM_CSn) and Write Enable (MEM_WRn) inputs low. Reading from the device is accomplished by taking the Chip Enable (SRAM_CSn) and the Output Enable (MEM_OEn) low while forcing Write Enable high. The contents of the memory location specified by the address pins will appear on the IO pins. Address space above 2000H is banked through the Configuration FPGA. The SRAM interface is shown in Figure 13.

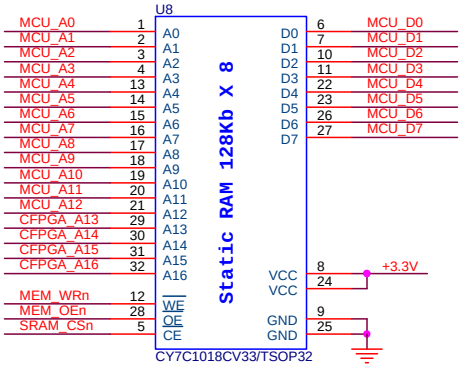


Figure 13 - MCU SRAM

3.1.3 MCU FLASH

Program memory is provided by the 1Mb x 8 FLASH (U6). To eliminate bus contention the device has separate Chip Enable (FLASH_CSn), Write Enable (MEM_WRn) and Output Enable (MEM_OEn) controls. Device programming occurs by executing the program command sequence. Address space above 2000H is banked through the Configuration FPGA. The FLASH interface is shown in Figure 14.

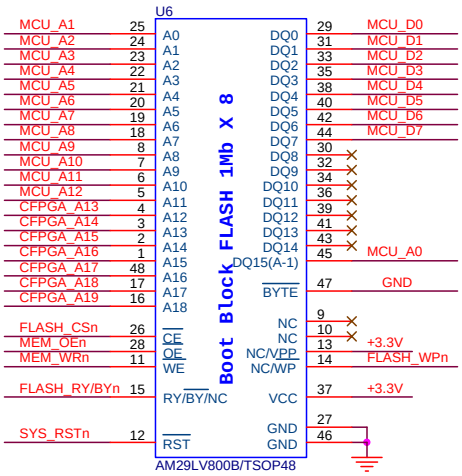


Figure 14 - MCU FLASH

3.1.4 MCU General Purpose IO (GPIO)

Header (P1) as shown in Figure 15 allows for connection to the unused MCU IO pins. The user can utilize this IO as required, e.g. external interrupts, external IO expansion etc. Note: The interface is LVTTL33 and the device is not 5V tolerant. (refer to the Cypress datasheet for CY7C68013 for more information)

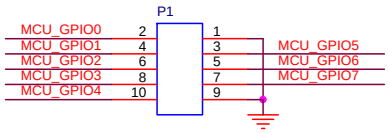


Figure 15 - MCU General Purpose IO Connector

3.1.5 MCU USB 2.0 Interface

Communication with the system is via the USB connector (J3), which interfaces directly with the MCU. The USB interface connector is a type B receptacle as shown in Figure 16. The CM1213 (U3/U4) provides ESD protection on the USB ports.

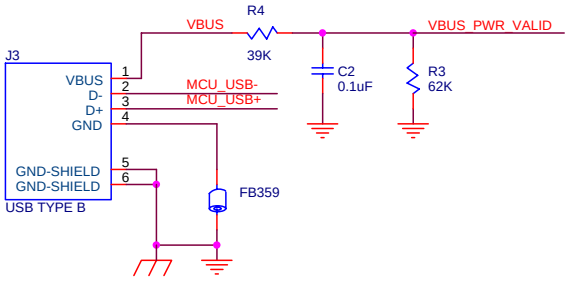


Figure 16 - USB Connector

3.1.6 RS232 Interface

An RS232 serial port (P2) is provided for low speed communication with the MCU. The RS-232 standard specifies output voltage levels between –5V to –15V for logical 1 and +5V to +15V for logical 0. Input must be compatible with voltages in the range of -3V to -15V for logical 1 and +3V to +15V for logical 0. This ensures data bits are read correctly even at maximum cable lengths between DTE and DCE, specified as 50 feet.

The RS-232 standard has two primary modes of operation, Data Terminal Equipment (DTE) and Data Communication Equipment (DCE). These can be thought of as host or PC for DTE and as peripheral for DCE. The DN6000K10 operates in the DCE mode only.

Figure 17 shows the implementation of the serial port on the DN6000K10

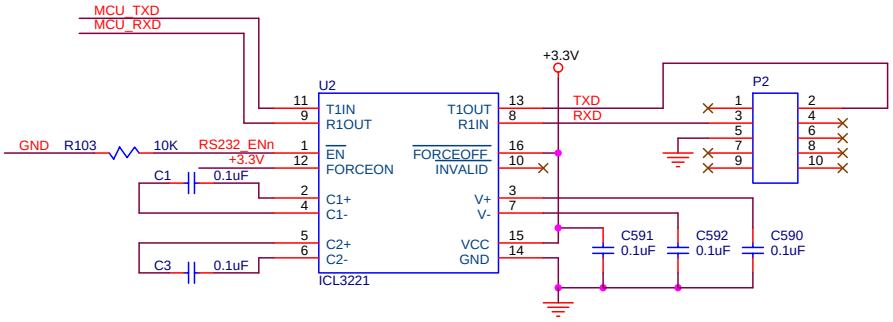


Figure 17 - MCU Serial Port

There are two signals attached to the MCU:

- Transmit Data
- Receive Data

TXD and RXD provide bi-directional transmission of transmit and receive data. No hardware handshaking is supported.

3.2 Configuration FPGA

The Xilinx Spartan-II XC2S150 (U13) is needed to handle the counters and state machines associated with the high-speed USB interface and the SmartMedia card. The FPGA contains 150K logic gates, 48K of BlockRAM and 260 user I/O's. The Verilog source code for the Configuration FPGA (ConfigFPGA.v) is provided on the CD-ROM.

The Configuration FPGA performs the following functions:

- Interface to the Micro Controller

  - Data Bus: MCU_D[0..7]
  - Address Signals: MCU_A[0..15]
  - Control Signals: MCU_RDn, MCU_WRn, MCU_CSn, MCU_OEn, MCU_PSENn
  - Clock: MCU_CLK
  - High Speed USB: SM_D[0..7], GPIF_RDYn, GPIF_CTL
- Interface to the SmartMedia
  - Data Bus: SM_D[0..7]
  - Control Signals: SM_REn, SM_WEn, SM_ALE, SM_CLE, SM_CEn, SM_RDYBUSYn
- Banked Address to the SRAM/FLASH
  - Upper Address Signals: CFPGA_A[13..19]
- FPGA Configuration, Serial/SelectMap
  - Data Bus for FPGA A,B,C: FPGA_1D[0..7]
  - Data Bus for FPGA D,E,F: FPGA_2D[0..7]
  - Data Bus for FPGA G,H,I: FPGA_3D[0..7]
  - Control Signals: FPGA_INIT_A, FPGA_DONE_A, FPGA_PROGn_A, FPGA_RD/WRn_A, FPGA_CSn_A, FPGA_BUSY_A, these signals are reproduced for FPGA A to FPGA I.
  - Clock: FPGA_DCLK
- FPGA Configuration, JTAG
  - JTAG Signals: FPGA_TCK, FPGA_TDI, FPGA_DONE/TDO, FPGA_TMS_ABC, FPGA_TMS_DEF, FPGA_TMS_GHI
- SRAM Chip Select Generation
  - Signal: SRAM_CSn
- FLASH Chip Select Generation
  - Signal: FLASH_CSn
- FPGA Configuration MODE Select DipSwitch
  - Signals: FPGA_MSEL[0..3]
- Interface to the UART Connectors
  - RS232 Signals from the FPGA's: PPCA_TXD, PPCA_RXD……PPCI_TXD, PPCI_RXD, for FPGA A to I.

  - RS232 Signals to the Connectors: PPC_TXD1, PPC_TXD2, PPC_TXD3, PPC_TXD4, PPC_RXD1, PPC_RXD2, PPC_RXD3, PPC_RXD4, PPC_MON1, PPC_MON2.
- LED Indicators
  - Signals: CFPGA_LEDn[0..3]
- GPIO between Configuration FPGA and FPGA's (9)
  - Signals: MB[1..40]

3.2.1 Configuration PROM/FPGA Programming

The Configuration FPGA (U13) is programmed using an in-system programmable configuration PROM (U12). The JTAG chain from the PROM is in a serial daisy chain with the Configuration FPGA, allowing simultaneous JTAG programming option of both devices. The Configuration FPGA is set to Master Serial Mode using dipswitch (S4). At power-up, the Configuration FPGA provides a configuration clock (CFPGA_CCLK) that drives the PROM. A short access time after CEn (CFPGA_DONE) and OE (CFPGA_INITn) are enabled, data is available on the PROM data (CFPGA_D0) pin that is connected to the Configuration FPGA. The programming header (J7) as shown in Figure 18, is used to download the files to the Configuration PROM/FPGA via a Xilinx Parallel IV cable.

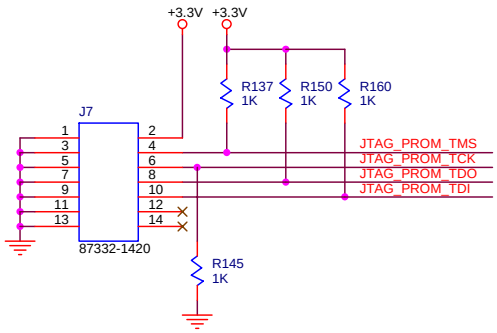


Figure 18 – Configuration PROM/FPGA Programming Header

3.2.2 Design Notes on the Configuration FPGA

Oscillator (X1) is a 48 MHz oscillator used to clock the Configuration FPGA. This part is soldered down to the PWB and is not intended to be user-configurable. The 48 MHz is divided down to 24 MHz in the Configuration FPGA to provide the clock for the micro controller (U7). The clock signal is labeled MCU_CLK on the schematic. The 48 MHz is used directly for the state machines in the Configuration FPGA for controlling the interface to the SmartMedia card. The maximum frequency for SelectMap configuration is 50 MHz without wait states.

Serial and JTAG configuration of the Virtex-II Pro FPGA's are back-off positions only. The 48 MHz clock can be divided down in the Configuration FPGA and used as a clock source to the PWB clock network (CFPGA_CLKOUT).

The signals MB[1..40] connects to the MB bus that links all the FPGA's. CFPGA_MSEL[0..2] selects the configuration mode of the Configuration FPGA (refer to Table 5) using dipswitch (S4).

Table 5 - FPGA Configuration Modes

Configuration Mode	M2	M1	M0	CLK Direction	Data Width	Serial Dout
Master Serial	0	0	0	Out	1	Yes
Slave Serial	1	1	1	In	1	Yes
Master SelectMAP	0	1	1	Out	8	No
Slave SelectMAP	1	1	0	In	8	No
Boundary Scan	1	0	1	N.A.	1	No

Note: Grayed options not supported by this design.

3.3 SmartMedia

The configuration bit file for the FPGA is copied to a SmartMedia card using the SmartDisk FlashPath Floppy Disk Adapter. The approximate file size for each possible FPGA option is shown below in Table 6. Note that several BIT files can be put on a 32MB card. The DN6000K10 is shipped with two 32-megabyte 3.3V SmartMedia cards. The DN6000K10 support card densities up to 128MB.

Note: Do **NOT** format the SmartMedia card using the default Windows file format program. Smart Media cards come pre-formatted from the factory, and files can be deleted from the card when they are no longer needed. If the SmartMedia card requires formatting, format the media with the program supplied by the FlashPath (SmartMedia floppy adapter) software.

Table 6 - FPGA configuration file sizes

Virtex-II Pro Device	Bitstream Length (bits)
XC2VP70	25,604,096
XC2VP100	33,645,312

SmartMedia Cards are available from www.computers4sure.com

3.3.1 SmartMedia Connector

Figure 19 shows J2, the SmartMedia connector used to download the configuration files to the FPGA.

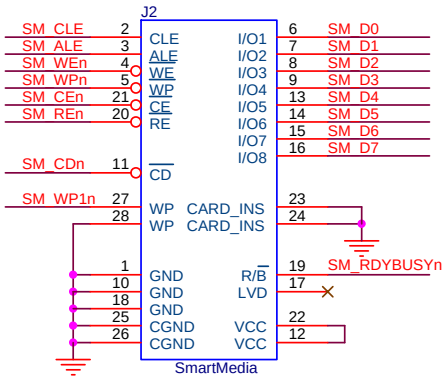


Figure 19 - SmartMedia Connector

Note: Do not press down on the top of the SmartMedia connector J2 if a SmartMedia card is not installed. The metal case shorts +3.3V to GND.

3.3.2 SmartMedia connection to Spartan (Configuration FPGA)/MCU

Table 7 shows the connection between the SmartMedia connector and the Configuration FPGA/MCU.

Table 7 - Connection between Configuration FPGA/MCU

Signal Name	Configuration FPGA/MCU	Connector
SM_D0	U13.K21	J2.6
SM_D1	U13.K22	J2.7
SM_D2	U13.J21	J2.8
SM_D3	U13.J20	J2.9
SM_D4	U13.J18	J2.13
SM_D5	U13.J22	J2.14
SM_D6	U13.J19	J2.15
SM_D7	U13.H19	J2.16
SM_CLE	U13.L20	J2.2

Signal Name	Configuration FPGA/MCU	Connector
SM_ALE	U13.L17	J2.3
SM_WEn	U13.L18	J2.4
SM_RDYBUSYn	U13.H18	J2.19
SM_CEn	U13.L21	J2.21
SM_REn	U13.L22	J2.20
SM_CDn	U7.106	J2.11
SM_WP1n	U7.82	J2.27

3.4 Boundary-Scan (JTAG, IEEE 1532) Mode

In boundary-scan mode, dedicated pins are used for configuring the Virtex-II Pro devices. The configuration is done entirely through the IEEE 1149.1 Test Access Port (TAP). The FPGA JTAG interfaces to IO on the Configuration FPGA. This allows manipulation of the data as required by the application and allows the JTAG chain to become an address on the existing bus. The processor can then read from, or write to the address representing the JTAG chain. FPGA's that are not populated requires feed through resistor to maintain the daisy chain connection between FPGA's.

3.4.1 FPGA JTAG Connector

Figure 20 shows J16, the JTAG connector used to download the configuration files to the FPGA's.

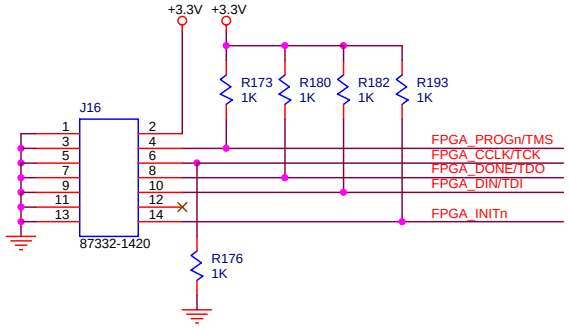


Figure 20 - FPGA JTAG Connector

3.4.2 FPGA JTAG connection to Configuration FPGA

Table 8 shows the connection between the FPGA JTAG connector and the Configuration FPGA.

Table 8 - FPGA JTAG connection to Configuration FPGA

Signal Name	Configuration FPGA	Connector
FPGA_CCLK/TCK	U13.N21	J16.6
FPGA_PROGn/TMS	U13.M20	J16.4
FPGA_DONE/TDO	U13.M19	J16.8
FPGA_DIN/TDI	U13.M18	J16.10
FPGA_INITn	U13.M22	J16.14

4 Clock Generation

4.1 Clock Methodology

The DN6000K10 Logic Emulation board has a flexible and configurable clocking scheme. Figure 21 is a block diagram showing the clocking resources and connections. FIXME

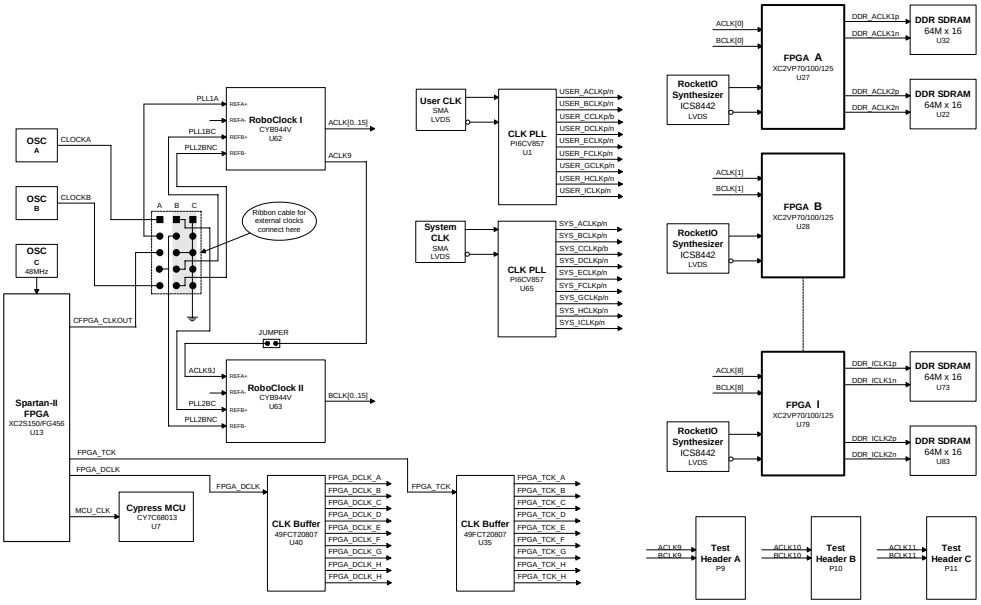


Figure 21 - Clocking Block Diagram

The clocking structures for the DN6000K10 include the following features:

- Two user-selectable socketed oscillators (X2, X3)
- One 48 MHz oscillator for the Configuration FPGA (X1)

- Two RoboclockII™ (CY7B994V) Multi-Phase PLL Clock Buffers
- External Differential User Clock Input (SMA Connectors J1/J4)
- System Oscillator (X4)
- Dedicated RocketIO Oscillators

The clock source selection grid formed by JP5, distributes clock signals (CLOCKA and CLOCKB) to two Roboclock PLL clock buffers (U62, U63). The clock outputs from the buffers are dispersed throughout the board. An external differential clock input option is available through the SMA connectors (J1, J4), which is the buffered and distributed throughout the board. A system oscillator (X4) is buffered and distributed throughout the board. This oscillator can be used to clock the Power PC's on each FPGA if required. Each FPGA has a dedicated RocketIO clock synthesizer driven by a 25MHz crystal. DDR clocks (DDR_CLKA….Ip/n) are generated by each individual FPGA. A dedicated 48MHz oscillator (X1) clocks the Configuration FPGA (U13), which in turn buffers the JTAG clock signal (FPGA_TCK) as well as the serial/parallel clock signal (FPGA_DCLK) required for FPGA configuration.

The connections between the FPGA's and various clocking resources are documented in Table 9, covering the clocking inputs and outputs, respectively.

Table 9 - Clocking inputs to the FPGA's

Signal Name	FPGA A Pin	Clock Refdes and Pin
ACLK0	U27.AU22	U62.89
BCLK0	U27.AN22	U63.89
USER_ACLKp	U27.K21	U1.3
USER_ACLKn	U27.J21	U1.2
SYS_ACLKp	U27.AP21	U65.3
SYS_ACLKn	U27.AN21	U65.2
RCKTIO_OSCT_Ap	U27.F21	U39.14
RCKTIO_OSCT_An	U27.G21	U39.15
RCKTIO_OSCB_Ap	U27.AT21	U39.11
RCKTIO_OSCB_An	U27.AU21	U39.12
DDR_ACLKp	U27.J22	U36. 5
DDR_ACLKn	U27.K22	U36.6
TST_HDRA_CLKIN	U27.AT22	P9.102

Signal Name	FPGA B Pin	Clock Refdes and Pin
ACLK1	U28.AU22	U62.91
BCLK1	U28.AN22	U63.91
USER_BCLKp	U28.K21	U1.5
USER_BCLKn	U28.J21	U1.6
SYS_BCLKp	U28.AP21	U65.5
SYS_BCLKn	U28.AN21	U65.6
RCKTIO_OSCT_Bp	U28.F21	U38.14
RCKTIO_OSCT_Bn	U28.G21	U38.15
RCKTIO_OSCB_Bp	U28.AT21	U38.11
RCKTIO_OSCB_Bn	U28,AU21	U38.12
Signal Name	**FPGA C Pin**	**Clock Refdes and Pin**
ACLK2	U29.J22	U62.94
BCLK2	U29.G22	U63.94
USER_CCLKp	U29.AP21	U1.10
USER_CCLKn	U29.AN21	U1.9
SYS_CCLKp	U29.K21	U65.10
SYS_CCLKn	U29.J21	U65.9
RCKTIO_OSCT_Cp	U29.F21	U37.14
RCKTIO_OSCT_Cn	U29.G21	U37.15
RCKTIO_OSCB_Cp	U29.AT21	U37.11
RCKTIO_OSCB_Cn	U29.AU21	U37.12
DDR_CCLKp	U29.AU22	U18. 5
DDR_CCLKn	U29.AT22	U18.6
Signal Name	**FPGA D Pin**	**Clock Refdes and Pin**
ACLK3	U53.K21	U62.96
BCLK3	U53.F21	U63.96
USER_DCLKp	U53.AN22	U1.20
USER_DCLKn	U53.AP22	U1.29
SYS_DCLKp	U53.J22	U65.20

SYS_DCLKn	AU53.K22	U65.19
RCKTIO_OSCT_Dp	U53.G22	U60.14
RCKTIO_OSCT_Dn	U53.F22	U60.15
RCKTIO_OSCB_Dp	U53.AU22	U60.11
RCKTIO_OSCB_Dn	U53.AT22	U60.12
DDR_DCLKp	U53.AT21	U36. 5
DDR_DCLKn	U53.AU22	U36.6
Signal Name	**FPGA E Pin**	**Clock Refdes and Pin**
ACLK4	U52.AU22	U62.66
BCLK4	U52.AN22	U63.66
USER_ECLKp	U52.K21	U1.22
USER_ECLKn	U52.J21	U1.23
SYS_ECLKp	U52.AP21	U65.22
SYS_ECLKn	U52.AN21	U65.23
RCKTIO_OSCT_Ep	U52.F21	U59.14
RCKTIO_OSCT_En	U52.G21	U59.15
RCKTIO_OSCB_Ep	U52.AT21	U59.11
RCKTIO_OSCB_En	U52.AU21	U59.12
Signal Name	**FPGA F Pin**	**Clock Refdes and Pin**
ACLK5	U51.AP21	U62.64
BCLK5	U51.AN21	U63.64
USER_FCLKp	U51.J22	U1.46
USER_FCLKn	U51.K22	U1.47
SYS_FCLKp	U51.AU22	U65.46
SYS_FCLKn	U51.AT22	U65.47
RCKTIO_OSCT_Fp	U51.G22	U50.14
RCKTIO_OSCT_Fn	U51.F22	U50.15
RCKTIO_OSCB_Fp	U51.AT21	U50.11
RCKTIO_OSCB_Fn	U51.AU21	U50.12
DDR_FCLKp	U51.K21	U41. 5

DDR_FCLKn	U51.J21	U41.6
Signal Name	**FPGA G Pin**	**Clock Refdes and Pin**
ACLK6	U80.AP21	U62.61
BCLK6	U80.AN21	U63.61
USER_GCLKp	U80.J22	U1.44
USER_GCLKn	U80.K22	U1.43
SYS_GCLKp	U80.AU22	U65.44
SYS_GCLKn	U80.AT22	U65.43
RCKTIO_OSCT_Gp	U80.G22	U70.14
RCKTIO_OSCT_Gn	U80.F22	U70.15
RCKTIO_OSCB_Gp	U80.AT21	U70.11
RCKTIO_OSCB_Gn	U80.AU21	U70.12
DDR_GCLKp	U80.K21	U87. 5
DDR_GCLKn	U80.J21	U87.6
Signal Name	**FPGA H Pin**	**Clock Refdes and Pin**
ACLK7	U78.AU22	U62.59
BCLK7	U78.AN22	U63.59
USER_HCLKp	U78.K21	U1.39
USER_HCLKn	U78.J21	U1.40
SYS_HCLKp	U78.AP21	U65.39
SYS_HCLKn	U78.AN21	U65.40
RCKTIO_OSCT_Hp	U78.G21	U67.14
RCKTIO_OSCT_Hn	U78.F21	U67.15
RCKTIO_OSCB_Hp	U78.AT21	U67.11
RCKTIO_OSCB_Hn	U78.AU21	U67.12
Signal Name	**FPGA I Pin**	**Clock Refdes and Pin**
ACLK8	U79.K21	U62.61
BCLK8	U79.F21	U63.61
USER_ICLKp	U79.AN22	U1.29
USER_ICLKn	U79.AP22	U1.30

SYS_ICLKp	U79.J22	U65.29
SYS_ICLKn	U79.K22	U65.30
RCKTIO_OSCT_Ip	U79.G22	U68.14
RCKTIO_OSCT_In	U79.F22	U68.15
RCKTIO_OSCB_Ip	U79.AU22	U68.11
RCKTIO_OSCB_In	U79.AT22	U68.12
DDR_ICLKp	U79.AT21	U66. 5
DDR_ICLKn	U79.AU21	U66.6

4.2 Clock Source Jumpers

The clock source grid JP5 gives the user the ability to select the clock input source to the RoboClock PLL buffers. A brief description of each pin is given in Table 10.

Table 10 - Clock Source Signals

Signal Name	Description	Connector
CFPGA_CLKOUT	Clock signal from the Configuration FPGA.	JP5.A3
CLOCKA	Clock signal from oscillator X3	JP5.A1
CLOCKB	Clock signal from oscillator X2	JP5.A5
PLL1B	Secondary clock input to RoboClock, differential pair with PLL1BN	JP5.B4
PLL1BN	Secondary clock input to RoboClock, differential pair with1 PLL1B	JP5.B5
PLL2B	Secondary clock input to RoboClock, differential pair with PLL2BN	JP5.B1
PLL2BN	Secondary clock input to RoboClock, differential pair with PLL2BN	JP5.B2
GND	Provides a ground reference for signals in the ribbon cable.	JP5.C1..C5

The PLL clock buffers can accept either LVTTL33 or Differential (LVPECL) reference inputs (refer to Figure 22).

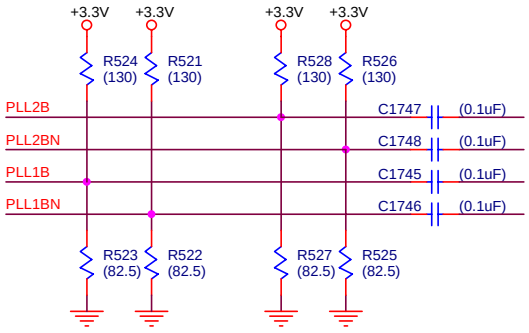


Figure 22 - LVPECL Clock Input and Termination

Note: The schematic shows capacitors in locations C1747, C1748, C1745, C1746. These are actually populated with 0-ohm resistors for direct connection to the RoboClock reference inputs. The terminating resistors to GDN and +3.3V are not stuffed. When using LVPECL, make the required hardware changes.

4.2.1 Clock Source Jumper Header

Figure 23 shows JP5, the clock source header connector used to select between different clock sources.

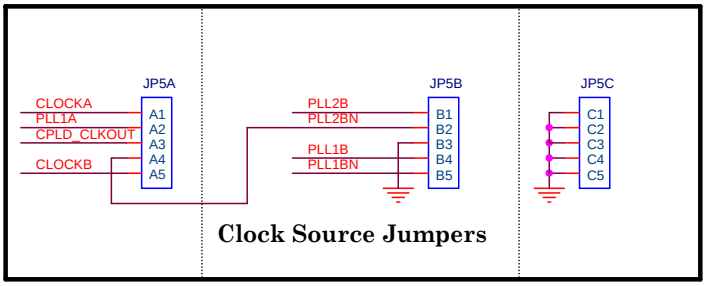


Figure 23 - Clock Source Jumper

4.3 Roboclocks

Two 3.3V half-can oscillator sockets (X2, X3) and the signal CFPGA_CLKOUT from the Configuration FPGA provide on-board input clock solutions. The DN6000K10 is shipped with both a 14.318MHz (X3) and a 33.33MHz (X2) oscillator. Neither X2 nor X3 are used by the configuration circuitry, so the user is free to stuff any standard 3.3 V half-can oscillators in the X2 and X3 positions. The oscillators interface to two high-speed multi-phase RoboClock buffers.

4.3.1 RoboClock PLL Clock Buffers

The CY7B994V (U62, U63) High-Speed Multi-Phase PLL Clock Buffers offer user-selectable control over system clock functions. Each chip has 16 output clocks along

with two feedback output clocks. Two sets of eight output clocks are jumper selectable for each chip. The feedback clocks are controlled separately.

Eighteen configurable outputs each drive terminated transmission lines with impedances as low as 50 while delivering minimal and specified output skews at LVTTL levels (refer to Figure 24). The outputs are arranged in five banks. Banks 1 to 4 of four outputs allow a divide function of 1 to 12, while simultaneously allowing phase adjustments in 625 ps - 1300 ps increments up to 10.4 ns. One of the output banks also includes an independent clock invert function. The feedback bank consists of two outputs, which allows divide-by functionality from 1 to 12 and limited phase adjustments. Any one of these eighteen outputs can be connected to the feedback input as well as driving other inputs.

Selectable reference input is a fault tolerance feature, which allows smooth change over to secondary clock source, when the primary clock source is not in operation. The reference inputs and feedback inputs are configurable to accommodate either LVTTL or Differential (LVPECL) inputs. The completely integrated PLL reduces jitter. Please refer to the datasheet for more detailed information.

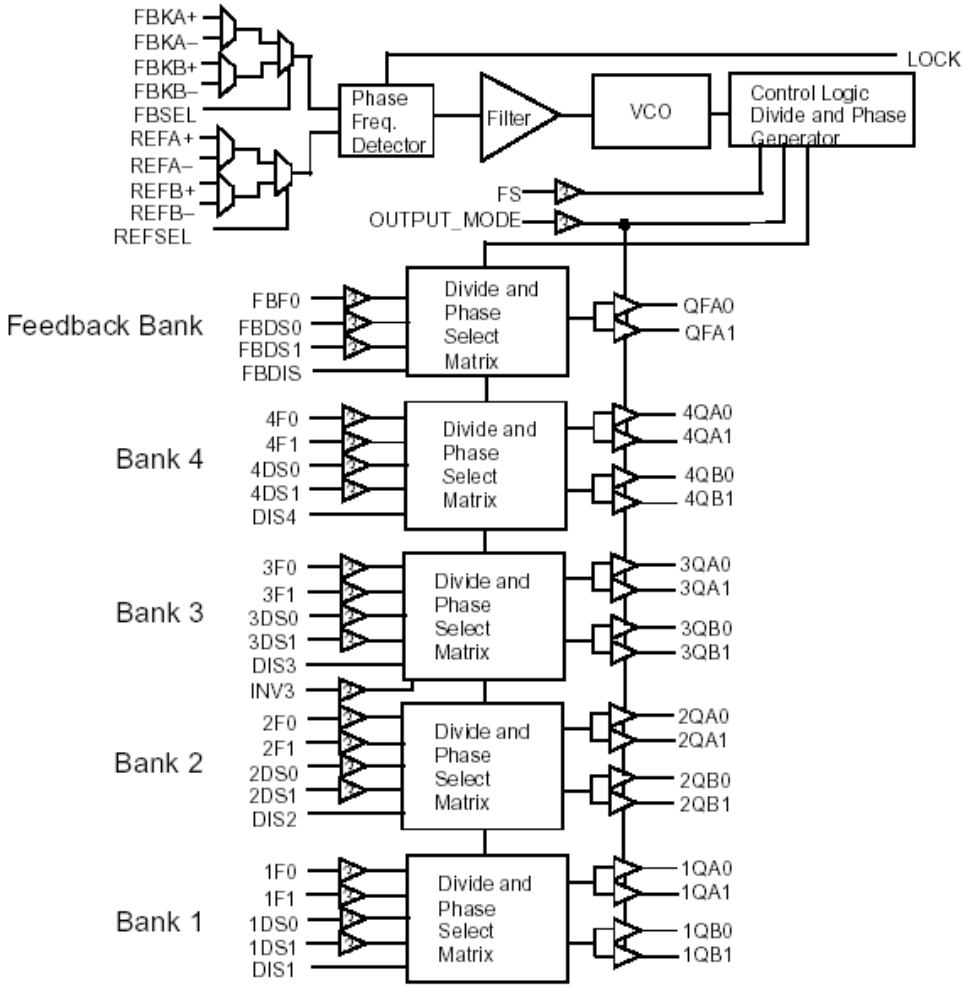


Figure 24 - RoboClock Functional Block Diagram

4.3.2 RoboClock Configuration Jumpers

Header JP4, JP6, and JP7 enable the user to configure the RoboClocks as required. These are 3-way headers and allow the signal to float (MID), or be pulled to GND (LOW) or +3.3V (HIGH). A brief description of each pin is given in Table 11.

Table 11 - RoboClock Configuration Signals

Signal Name	Description	Connector
ROBO1_REFSEL1	ROBOCLOCK #1, Reference Select Input: The REFSEL input controls how the reference input is configured. When LOW, it will use the REFA pair (PLL1A) as the reference input. When HIGH, it will use the REFB pair (PLL1BC, PLL1BNC) as the	JP4.B1

Signal Name	Description	Connector
	reference input. This input has an internal pull-down.	
ROBO1_FS	ROBOCLOCK #1, Frequency Select: This input must be set according to the nominal frequency (fNOM). Refer to Table 1 in the datasheet.	JP4.B2
ROBO1_FBF0	ROBOCLOCK #1, Feedback Output Phase Function Select: Controls the phase function of bank 3 & 4 (CCLK) of outputs, refer to Table 3 in the datasheet.	JP4.B3
ROBO1_FBDS0	ROBOCLOCK #1, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.	JP4.B4
ROBO1_FBDS1	ROBOCLOCK #1, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.	JP4.B5
ROBO1_FBDIS	ROBOCLOCK #1, Feedback Disable: This input controls the state of QFA[0:1]. When HIGH, the QFA[0:1] is disabled to the "HOLD-OFF" or "HI-Z" state; the disable state is determined by OUTPUT_MODE. When LOW, the QFA[0:1] is enabled. Refer to Table 5 in the datasheet. This input has an internal pull-down.	JP4.B6
ROBO1_F0	ROBOCLOCK #1, Output Phase Function Select: Controls the phase function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 3 in the datasheet.	JP4.B7
ROBO1_F1	ROBOCLOCK #1, Output Phase Function Select: Controls the phase function of bank 1, 2, 3 & 4 (DCLK) of outputs. Refer to Table 3 in the datasheet.	JP4.B8
ROBO1_DS0	ROBOCLOCK #1, Output Divider Function Select: Controls the divider function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 4 in the datasheet.	JP4.B9
ROBO1_DS1	ROBOCLOCK #1, Output Divider Function	JP4.B10

Signal Name	Description	Connector
	Select: Controls the divider function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 4 in the datasheet.	
ROBO2_REFSEL1	ROBOCLOCK #2, Reference Select Input: The REFSEL input controls how the reference input is configured. When LOW, it will use the REFA pair (PLL1A) as the reference input. When HIGH, it will use the REFB pair (PLL1BC, PLL1BNC) as the reference input. This input has an internal pull-down.	JP7.B1
ROBO2_FS	ROBOCLOCK #2, Frequency Select: This input must be set according to the nominal frequency (fNOM). Refer to Table 1 in the datasheet.	JP7.B2
ROBO2_FBF0	ROBOCLOCK #2, Feedback Output Phase Function Select: Controls the phase function of bank 3 & 4 (CCLK) of outputs, refer to Table 3 in the datasheet.	JP7.B3
ROBO2_FBDS0	ROBOCLOCK #2, Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.	JP7.B4
ROBO2_FBDS1	ROBOCLOCK #12 Feedback Divider Function Select: These inputs determine the function of the QFA0 and QFA1 outputs. Refer to Table 4 in the datasheet.	JP7.B5
ROBO2_FBDIS	ROBOCLOCK #2, Feedback Disable: This input controls the state of QFA[0:1]. When HIGH, the QFA[0:1] is disabled to the "HOLD-OFF" or "HI-Z" state; the disable state is determined by OUTPUT_MODE. When LOW, the QFA[0:1] is enabled. Refer to Table 5 in the datasheet. This input has an internal pull-down.	JP7.B6
ROBO2_F0	ROBOCLOCK #2, Output Phase Function Select: Controls the phase function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 3 in the datasheet.	JP7.B7

Signal Name	Description	Connector
ROBO2_F1	ROBOCLOCK #2, Output Phase Function Select: Controls the phase function of bank 1, 2, 3 & 4 (DCLK) of outputs. Refer to Table 3 in the datasheet.	JP7.B8
ROBO2_DS0	ROBOCLOCK #2, Output Divider Function Select: Controls the divider function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 4 in the datasheet.	JP7.B9
ROBO2_DS1	ROBOCLOCK #2, Output Divider Function Select: Controls the divider function of bank 1, 2, 3 & 4 (ACLK) of outputs. Refer to Table 4 in the datasheet.	JP7.B10
OSCA	Enable for Oscillator A (X9)	JP6.B1
OSCB	Enable for Oscillator B (X8)	JP6.B1
ROBO1_DIS	ROBOCLOCK #1, Output Disable: Each input controls the state of the respective output bank. When HIGH, the output bank is disabled to the "HOLD-OFF" or "HI-Z" state; the disable state is determined by OUTPUT_MODE. When LOW, the [1:4]Q[A:B][0:1] is enabled. (See *Table 5in Datasheet).* These inputs each have an internal pull-down.	JP6.B5
ROBO2_DIS	ROBOCLOCK #2, Output Disable: Each input controls the state of the respective output bank. When HIGH, the output bank is disabled to the "HOLD-OFF" or "HI-Z" state; the disable state is determined by OUTPUT_MODE. When LOW, the [1:4]Q[A:B][0:1] is enabled. (See *Table 5in Datasheet).* These inputs each have an internal pull-down.	JP6.B6
ROBO1_MODE	ROBOCLOCK #1, Output Mode: This pin determines the clock outputs' disable state. When this input is HIGH, the clock outputs will disable to high-impedance (HI-Z). When this input is LOW, the clock outputs will disable to "HOLD-OFF" mode. When in MID, the device will enter factory test mode.	JP6.B7

Signal Name	Description	Connector
ROBO2_MODE	ROBOCLOCK #2, Output Mode: This pin determines the clock outputs' disable state. When this input is HIGH, the clock outputs will disable to high-impedance (HI-Z). When this input is LOW, the clock outputs will disable to "HOLD-OFF" mode. When in MID, the device will enter factory test mode.	JP6.B8

4.3.3 Roboclock Configuration Headers

Figure 25 shows JP4, JP6, and JP7, the RoboClock configuration headers.

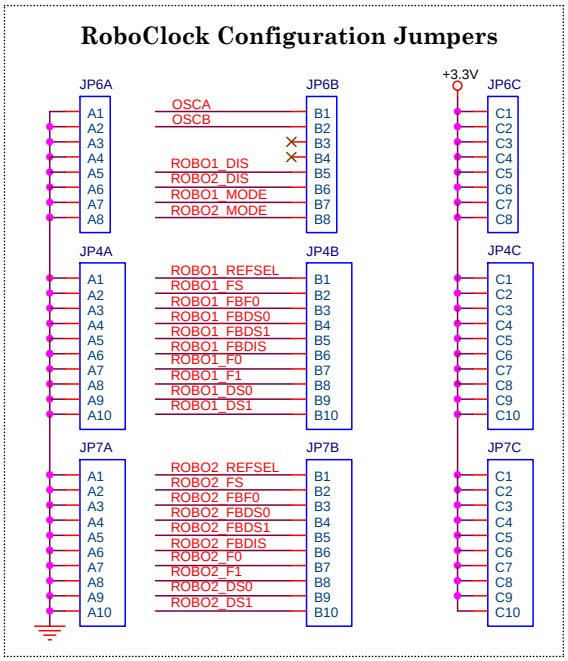


Figure 25 - RoboClock Configuration Jumpers

4.3.4 Useful Notes and Hints

The RoboClock consistently outputs ~32.5MHz signals in cases of improper settings or unacceptable clock inputs. This was observed when the CY7B994V part was operating at a nominal frequency f_{NOM} of 36.4MHz with FS set LOW. Identical clocks were sent to PLL2B and PLL2BN.

For the CY7B994V part, the operating frequency can reach up to 200 MHz. However, the maximum output frequency is 185MHz. This means when 185 MHz < f_{NOM} < 200MHz, the output divider must be set to at least 2. Otherwise, the RoboClocks will output garbage.

4.3.5 Customizing the Oscillators

The user can customize the frequency of the clock networks by stuffing different oscillators in X2 and X3. The DN6000K10 is shipped with a 14.318MHz oscillator in location X3 and a 33.333MHz oscillator in X2. The RoboClocks are not +5V tolerant, so +3.3V oscillators are necessary.

The Dini Group suggests Digi-Key (http://www.digikey.com/) as a possible source for the oscillators. Of note is the Epson line of oscillators called the SG-8002 Programmable Oscillators. Any frequency between 1.00MHz–106.25MHz can be procured in the normal Digi-Key shipping time of 24 hours. A half-can, +3.3 V CMOS version is needed with a tolerance of 50ppm. The part number for an acceptable oscillator from this family would be:

SG-8002DC-PCB-ND

- Package SG-8002DC (Halfcan)
- Output Enable
- 3.3 V CMOS
- ±±50 ppm

If the order is placed via the web page, the requested frequency to two decimal places is placed in the Web Order Notes. The datasheet is on the CD-ROM for this oscillator. Any polarity of output enabled for each oscillator (on pin 1) is acceptable. Ensure the proper jumper settings for JP6.B1/JP6.B2. See Table 11 for a description.

4.3.6 Common Clock Source Selections

The following configuration is the most common:

Configuration 1: CLOCKA → PLL1A, CLOCKB → PLL2BN

RoboClock #1 (U62) is driven from oscillator X3. RoboClock #2 (U63) is driven from oscillator X2. RoboClock #2 can also be driven from RoboClock #1 output (ACLK9) if required.

4.4 External Clocks

The clock source jumper (JP5) allows the user a simple means to attach external clocks to the clock grid. The user can attach 10-pin ribbon cable to JP5B/C, which allows for connection the differential pair inputs of both RoboClocks. JP5C ground pins for signal integrity. These signals are described in Table 10. Both differential pairs provide some flexibility. The user can bring a single 3.3V TTL input. It can be attached to either input. However, the other input must be left open. The user can provide a differential clock input to the pair to the RoboClocks.

4.4.1 External SMA Clock

J1/J4 are SMA connectors to allow an external differential clock (USER_CLKp/n) input to all the FPGA's via a PLL clock driver (U1). Resistors (R100, R116) allows for AC coupling if required. Refer Figure 26.

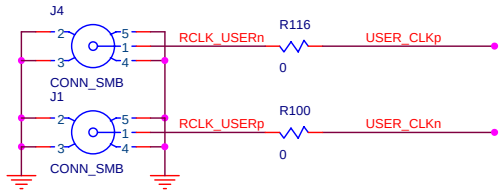


Figure 26 - External SMA Clock

4.4.2 Connections between FPGA's and External SMA Clock Buffer

The connection between the FPGA's and the external SMA clock buffer are shown in Table 12.

Table 12 - Connection between FPGA and External PPC Oscillator

Signal Name	FPGA Pin	External SMA Clock Buffer
USER_ACLKp	U27.K21	U1.3
USER_ACLKn	U27.J21	U1.2
USER_BCLKp	U28.K21	U1.5
USER_BCLKn	U28.J21	U1.6
USER_CCLKp	U29.AT21	U1.10
USER_CCLKn	U29.AU21	U1.9
USER_DCLKp	U53.AU22	U1.20
USER_DCLKn	U53.AT22	U1.19
USER_ECLKp	U52.K21	U1.22
USER_ECLKn	U52.J21	U1.23
USER_FCLKp	U51.J22	U1.46
USER_FCLKn	U51.K22	U1.47
USER_GCLKp	U80.J22	U1.44
USER_GCLKn	U80.K22	U1.43
USER_HCLKp	U78.K21	U1.39
USER_HCLKn	U78.J21	U1.40
USER_ICLKp	U79.AU22	U1.29

Signal Name	FPGA Pin	External SMA Clock Buffer
USER_ICLKn	U79.AT22	U1.30

4.5 DDR Clocking

The DDR Clock is generated in the FPGA by using the Digital Clock Managers (DCM). Clocking for DDR SDRAM requires the transmission of two clocks, the positive clock and the negative clock, SSTL_2 differential. These two clocks are 180° out of phase from each other, and their phase alignment must be tightly controlled. In order to prevent signal integrity problems and timing differences from becoming an issue, it is preferable for each device, whether memory or register, to have its own clock.

While it is possible for each device to have a positive and negative clock generated by the FPGA, this unnecessarily consumes pins that could be used elsewhere. To save these pins, an externally DDR SDRAM clock driver is used. The clock is routed to the DDR PLL Clock Driver that distributes the individual clocks to the separate DDR devices.

4.5.1 Clocking Methodology

This section describes the DDR clocking methodology implemented in the reference design (refer to Figure 27). The first DCM generates CLK0 and CLK90. CLK0 directly follows the user-supplied input clock (one of the clock sources, ACLK, BCLK etc.). This DCM also supplies the CLKDV output, which is the input clock divided by 16 used for the AUTO REFRESH counter. The second DCM in the controller block (DCM2_RECAPTURE) generates a phase-shifted version of the user input clock. It is used to recapture data from the DQS clock domain during a memory Read. Data recaptured in the rclk domain is then transferred to the system clock domain. The phase-shift value is specific to the system and must be programmed accordingly.

When adequate DCM resources are available, a third DCM can be used for better timing margins. This DCM is used to generate WCLK, a phase shifted version of the system clock. WCLK is used to clock data at the DDR IOB registers during a Write.

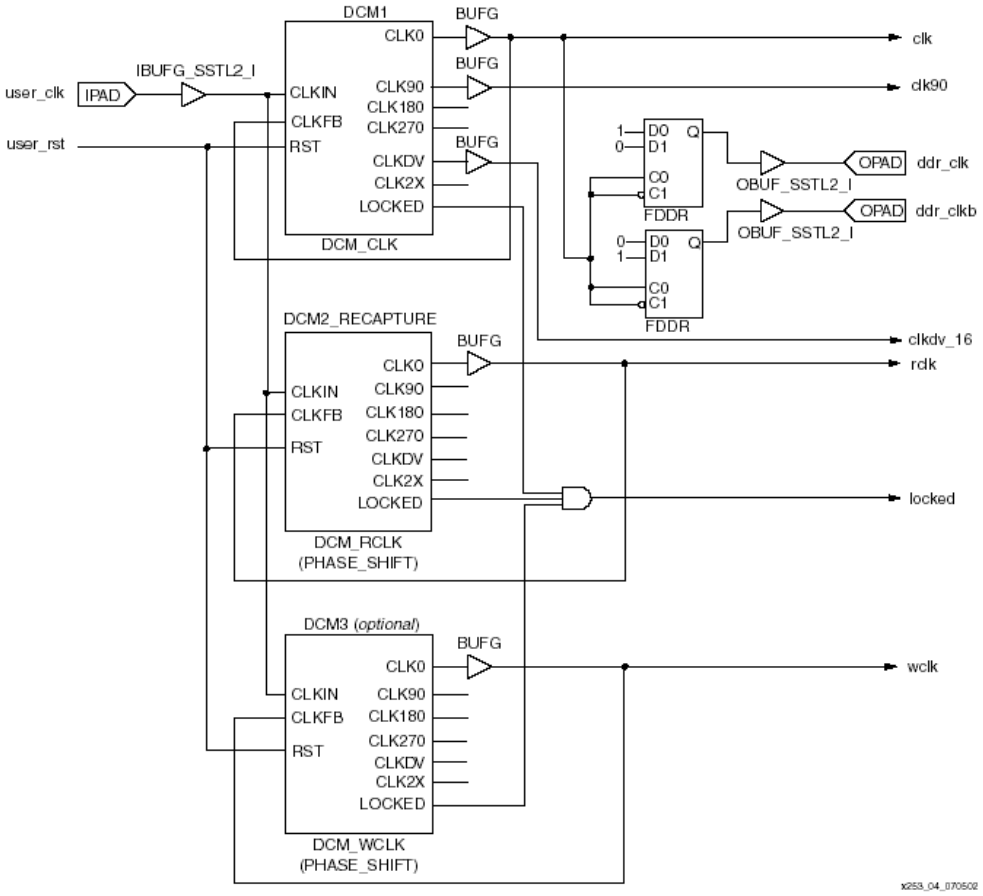


Figure 27 - DDR DCM Implementation

4.5.2 Connections between FPGA's and DDR PLL Clock Buffer

The connection between the FPGA's and the DDR PLL Clock Drivers consists of SSTL_2 differential pairs. A feedback reference clock input is provided from the PLL clock driver to each FPGA. The connections for all the FPGA's are shown in Table 13.

Table 13 - Connection between FPGA's and DDR PLL Clock Drivers

Signal Name	FPGA Pin	DDR PLL Clock Driver
DDR_ACLKp	U27.J22	U36.5
DDR_ACLKn	U27.K22	U36.6
DDR_CCLKp	U29.AU22	U18.5
DDR_CCLKn	U29.AT22	U18.6
DDR_DCLKp	U53.AU22	U61.5

DDR_DCLKn	U53.AT22	U61.6
DDR_FCLKp	U51.J22	U41.5
DDR_FCLKn	U51.K22	U41.6
DDR_GCLKp	U80.J22	U87.5
DDR_GCLKn	U80.K22	U87.6
DDR_ICLKp	U79.AU22	U66.5
DDR_ICLKn	U79.AT22	U66.6

4.6 Power PC (PPC) Clock – Sytem Clock

A 3.3 V half-can oscillator (X4), and the signal SYS_CLK provide an external clock source for the PPC. The oscillator is socketed and the DN6000K10 is shipped with a 100MHz oscillator, refer to Figure 28.

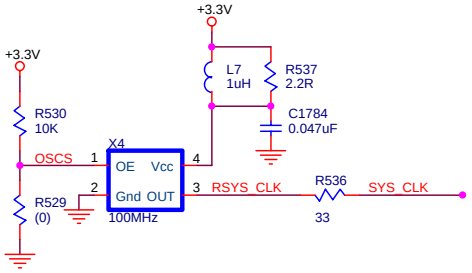


Figure 28 - PPC External Clock

4.6.1 Clocking Methodology

Refer to the Xilinx application notes for more information on this subject.

4.6.2 Connections between FPGA's and System Clock Buffer

The connection between the FPGA's and the external oscillator buffer are shown in Table 14.

Table 14 - Connection between FPGA and External PPC Oscillator

Signal Name	FPGA Pin	DDR PLL Clock Driver
SYS_ACLKp	U27.AT21	U65.3
SYS_ACLKn	U27.AU21	U65.2
SYS_BCLKp	U28.AT21	U65.5
SYS_BCLKn	U28.AU21	U65.6
SYS_CCLKp	U29.K21	U65.10

Signal Name	FPGA Pin	DDR PLL Clock Driver
SYS_CCLKn	U29.J21	U65.9
SYS_DCLKp	U53.J22	U65.20
SYS_DCLKn	U53.K22	U65.19
SYS_ECLKp	U52.AT21	U65.22
SYS_ECLKn	U52.AU21	U65.23
SYS_FCLKp	U51.AU22	U65.46
SYS_FCLKn	U51.AT22	U65.47
SYS_GCLKp	U80.AU22	U65.44
SYS_GCLKn	U80.AT22	U65.43
SYS_HCLKp	U78.AT21	U65.39
SYS_HCLKn	U78.AU21	U65.40
SYS_ICLKp	U79.J22	U65.29
SYS_ICLKn	U79.K22	U65.30

4.7 Rocket IO Programmable Clocks

The DN6000K10 provides one crystal oscillator-to-differential LVDS frequency syntheszer per FPGA. These frequency syntheszer are serially programmable. The use of this variable clock source, allows designers to prototype various interconnect technologies with different clock source requirements. . The dual output LVDS clocks are routed to the top and bottom RocketIO reference clock inputs. The PLL architecture for the RocketIO transceivers uses the reference clock as the interpolation source to clock the serial data. Removing the reference clock will stop the RX and TX PLLs from working. Therefore, a reference clock must be provided at all times. The serial transceiver input is locked to the input data stream through Clock and Data Recovery (CDR), a built in feature of the RodketIO transceiver. There are eight clock inputs into each RocketIO transceiver instantiation. REFCLK and BREFCLK are reference clocks generated from an external sources and presented to the FPGA as differential inputs. The reference clocks connect to the REFCLK or BREFCLK ports of the RocketIO multi-gigabit transceiver (MGT). While only one of these reference clocks is needed to drive the MGT, BREFCLK or BREFCLK2 must be used for serial speeds of 2.5 Gbps or greater. The reference clock also locks a Digital Clock Manager (DCM) or a BUFG to generate all of the other clocks for the GT. *Never run a reference clock through a DCM, since unwanted jitter will be introduced.*

4.7.1 Clocking Methodology

At speeds of 2.5 Gbps or greater, REFCLK configuration introduces more than the maximum allowable jitter to the RocketIO transceiver. For these higher speeds, BREFCLK configuration is required. The BREFCLK configuration uses dedicated routing resources that reduce jitter. BREFCLK must enter the FPGA through dedicated clock I/O. BREFCLK can connect to the BREFCLK inputs of the transceiver and the CLKIN input of the DCM for creation of USRCLKs. For more information refer to the Rocket IO User Guide available from the Xilinx website.

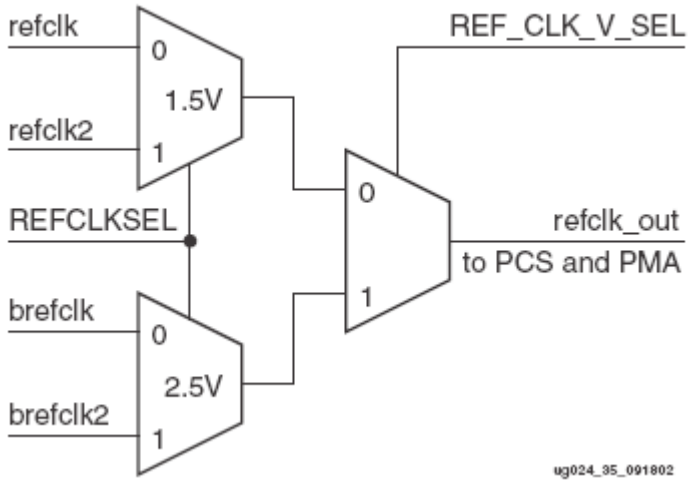


Figure 29 - REFCLK/BREFCLK Selection Logic

4.7.2 ICS8442 Programmable LVDS Clock Synthesizer

The DN6000K10 uses the ICS8442 LVDS clock synthesizer for generating various clock frequencies:

- VCO range: 250MHz to 700MHZ
- Output Frequency range: 31.25MHz to 700MHz
- RMS period jitter: 2.7ps (typical)
- Cycle-to-cycle jitter: 18ps (typical)

Please refer to the manufacturers datasheet for more information
http://www.icst.com/

4.7.3 Connections between FPGA's and RocketIO Clock Synthesizers

The connection between the FPGA's and the RocketIO clock synthesizers are shown in Table 15.

Table 15 - Connections between FPGA's and Rocket IO Clock Synthesizers

Signal Name	FPGA Pin	OSCILLATOR
RCKTIO_OSCT_Ap	U39.14	U27.F21
RCKTIO_OSCT_An	U39.15	U27.G21
RCKTIO_OSCB_Ap	U39.11	U27.AT21
RCKTIO_OSCB_An	U39.12	U27.AU21
RCKTIO_OSCT_Bp	U38.14	U28.F21
RCKTIO_OSCT_Bn	U38.15	U28.G21
RCKTIO_OSCB_Bp	U38.11	U28.AT21
RCKTIO_OSCB_Bn	U38.12	U28.AU21
RCKTIO_OSCT_Cp	U37.14	U29.F21
RCKTIO_OSCT_Cn	U37.15	U29.G21
RCKTIO_OSCB_Cp	U37.11	U29.AT21
RCKTIO_OSCB_Cn	U37.12	U29.AU21
RCKTIO_OSCT_Dp	U60.14	U53.G22
RCKTIO_OSCT_Dn	U60.15	U53.F22
RCKTIO_OSCB_Dp	U60.11	U53.AU22
RCKTIO_OSCB_Dn	U60.12	U53.AT22
RCKTIO_OSCT_Ep	U59.14	U52.F21
RCKTIO_OSCT_En	U59.15	U52.G21
RCKTIO_OSCB_Ep	U59.11	U52.AT21
RCKTIO_OSCB_En	U59.12	U52.AU21
RCKTIO_OSCT_Fp	U50.14	U51.G22
RCKTIO_OSCT_Fn	U50.15	U51.F22
RCKTIO_OSCB_Fp	U50.11	U51.AT21
RCKTIO_OSCB_Fn	U50.12	U51.AU21
RCKTIO_OSCT_Gp	U70.14	U80.G22
RCKTIO_OSCT_Gn	U70.15	U80.F22
RCKTIO_OSCB_Gp	U70.11	U80.AT21

Signal Name	FPGA Pin	OSCILLATOR
RCKTIO_OSCB_Gn	U70.12	U80.AU21
RCKTIO_OSCT_Hp	U67.14	U78.F21
RCKTIO_OSCT_Hn	U67.15	U78.G21
RCKTIO_OSCB_Hp	U67.11	U78.AT21
RCKTIO_OSCB_Hn	U67.12	U78.AU21
RCKTIO_OSCT_Ip	U68.14	U78.G22
RCKTIO_OSCT_In	U68.15	U78.F22
RCKTIO_OSCB_Ip	U68.11	U78.AU22
RCKTIO_OSCB_In	U68.12	U78.AT22

5 Reset Topology

5.1 DN6000K10 Reset

The voltage monitor device from Linear Technology, P/N LTC2900 (U5), allows a push-button reset function that is used to reset the DN6000K10. Figure 30 shows the distribution of the reset signal SYS_RSTn. In addition to controlling the reset, the power supplies rails +1.5V, +2.5V, +3.3V, and +5V are monitored for under-voltage conditions, that will cause the assertion of the SYS_RSTn signal. LED DS2.2 when lit, means that reset is asserted, refer the section describing the GPIO LED's.

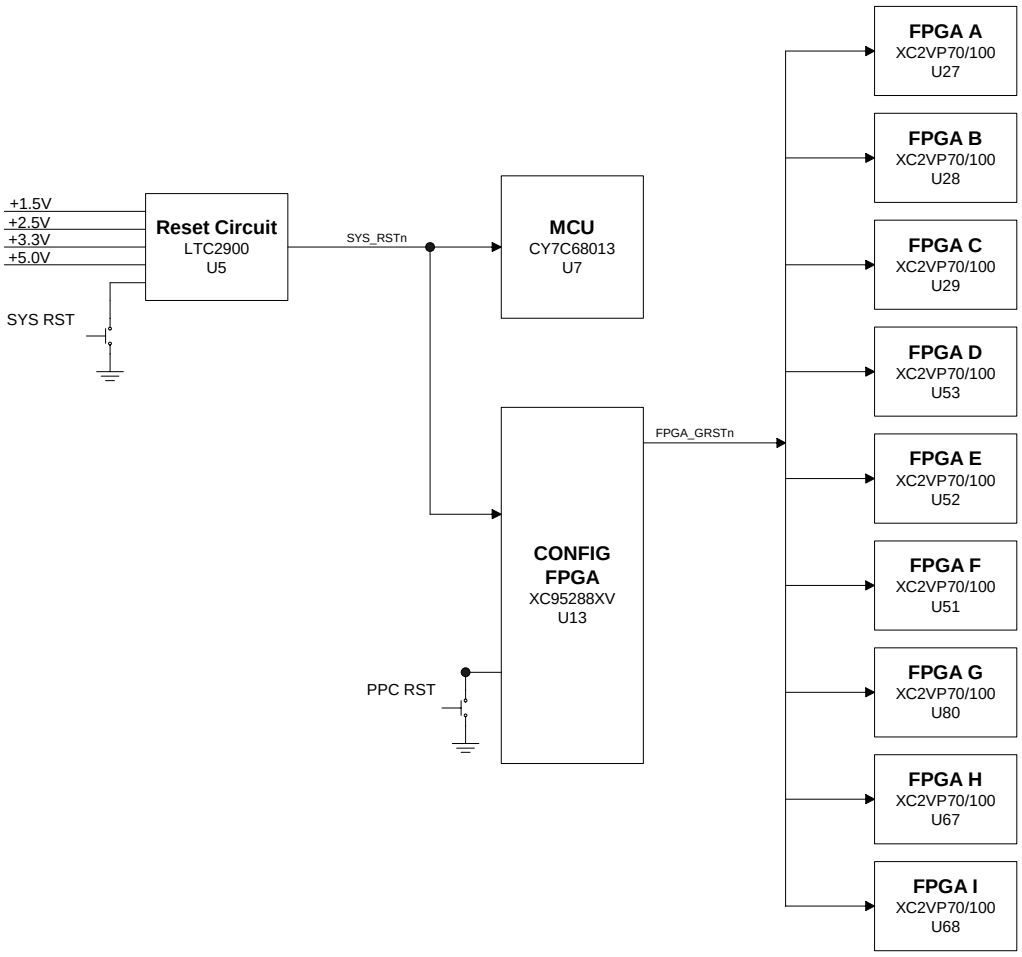


Figure 30 - Reset Topology Block Diagram

Depressing the reset push-button (S1) causes the following sequence of events:

1. Reset of the Configuration FPGA and MCU
2. Reset of FPGA's through FPGA_GRSTn signal
3. FPGA configuration is cleared
4. If the dipswitch is set for SelectMAP configuration option, and there is a valid SmartMedia card inserted into the socket, then the FPGA's will be configured. A SmartMedia card is valid if it complies with the SSFDC specification and contains a file named "main.txt" in the root directory. If the card is invalid or there is no card present, then the FPGA will not be configured.
5. The Main Menu will appear in the Terminal Window.

Note: The identical sequence of events occurs at power-up.

5.2 PPC Reset

The DN6000K10 also contains another RESET push-button (S3) used to reset the PPC's in each FPGA. This signal is pulled up on the DN6000K10. The user is responsible for debouncing the reset signal in the Configuration FPGA. One of the MB[1..40] signals must be used to reset the PPC's in the FPGA's. Table 16 shows the connection between the reset push-button and the FPGA.

Table 16 - PPC Reset

Signal Name	FPGA Pin	Push-Button Switch
PPC_RESETn	U13.H5	S3.4

6 Memory

The DN6000K10 provides two different memory technologies to the user. FLASH and DDR SDRAM in various densities.

6.1 FLASH

The FLASH memory components on the DN6000K10 can accommodate up to 4M x 16 devices, refer to Figure 31 as an example of a FLASH interface (shown is the FLASH device on FPGA A). In addition to programming the FPGA and storing bitstreams, the FLASH may be used for non-volatile storage.

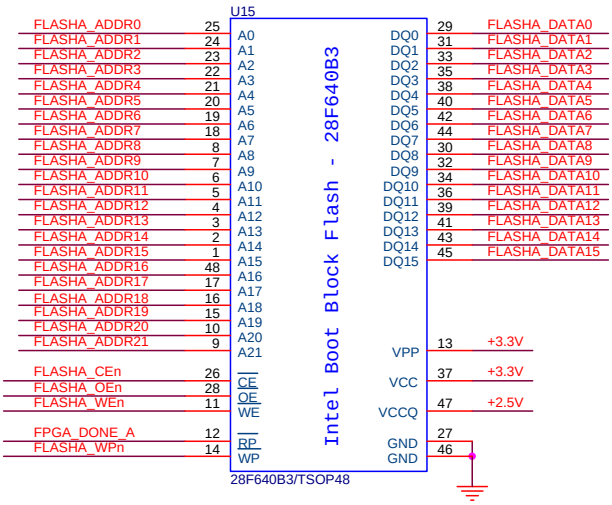


Figure 31 - FLASH Connection

The Intel Advanced Boot Block Flash Memory (C3) device, supports read-array mode operations at various IO voltages (1.8V and 3V) and erase and program operations at 3V or 12V VPP. On the DN6000K10C, VPP is 3.3V. The DN6000K10C interfaces to the FLASH at +2.5V levels.

This family of devices is capable of fast programming at 12V (not utilized on the DN6000K10C). The C3 device features the following:

- Enhanced blocking for easy segmentation of code and data or additional design flexibility
- Program Suspend to Read command
- VCCQ input of 1.65V–2.5V or 2.7V–3.6V on all I/Os
- Maximum program and erase time specification for improved data storage

For more information on this part please refer to the Intel P/N TE28F640C3TC80 datasheet.

6.1.1 FLASH Connection to the FPGA's

The FLASH memory components are connected as listed in Table 17. The VCCO of the IO banks are connected to +2.5V.

Table 17 - Connection between FPGA and FLASH

Signal Name	FPGA Pin	FLASH
FLASHA_ADDR0	U27.P40	U15.25
FLASHA_ADDR1	U27.N38	U15.24
FLASHA_ADDR2	U27.N39	U15.23
FLASHA_ADDR3	U27.N40	U15.22
FLASHA_ADDR4	U27.M38	U15.21
FLASHA_ADDR5	U27.M39	U15.20
FLASHA_ADDR6	U27.M40	U15.19
FLASHA_ADDR7	U27.L38	U15.18
FLASHA_ADDR8	U27.H40	U15.8
FLASHA_ADDR9	U27.G38	U15.7
FLASHA_ADDR10	U27.G39	U15.6
FLASHA_ADDR11	U27.G40	U15.5
FLASHA_ADDR12	U27.F39	U15.4
FLASHA_ADDR13	U27.F40	U15.3
FLASHA_ADDR14	U27.E40	U15.2

Signal Name	FPGA Pin	FLASH
FLASHA_ADDR15	U27.D40	U15.1
FLASHA_ADDR16	U27.AA37	U15.48
FLASHA_ADDR17	U27.L39	U15.17
FLASHA_ADDR18	U27.L40	U15.16
FLASHA_ADDR19	U27.K38	U15.15
FLASHA_ADDR20	U27.H38	U15.10
FLASHA_ADDR21	U27.H39	U15.9
FLASHA_DATA0	U27.R38	U15.29
FLASHA_DATA1	U27.T39	U15.31
FLASHA_DATA2	U27.U40	U15.33
FLASHA_DATA3	U27.U38	U15.35
FLASHA_DATA4	U27.V38	U15.38
FLASHA_DATA5	U27.W39	U15.40
FLASHA_DATA6	U27.Y40	U15.42
FLASHA_DATA7	U27.AA40	U15.44
FLASHA_DATA8	U27.T40	U15.30
FLASHA_DATA9	U27.T38	U15.32
FLASHA_DATA10	U27.U39	U15.34
FLASHA_DATA11	U27.V39	U15.36
FLASHA_DATA12	U27.W40	U15.39
FLASHA_DATA13	U27.W38	U15.41
FLASHA_DATA14	U27.Y39	U15.43
FLASHA_DATA15	U27.AA39	U15.45
FLASHA_CEN	U27.P38	U15.26
FLASHA_OEN	U27.R40	U15.28
FLASHA_WEN	U27.J39	U15.11
FLASHA_WPN	U27.K40	U15.14
FLASHB_ADDR0	U28.U42	U16.25
FLASHB_ADDR1	U28.T41	U16.24

Signal Name	FPGA Pin	FLASH
FLASHB_ADDR2	U28.R41	U16.23
FLASHB_ADDR3	U28.R42	U16.22
FLASHB_ADDR4	U28.P41	U16.21
FLASHB_ADDR5	U28.P42	U16.20
FLASHB_ADDR6	U28.N41	U16.19
FLASHB_ADDR7	U28.N42	U16.18
FLASHB_ADDR8	U28.G41	U16.8
FLASHB_ADDR9	U28.G42	U16.7
FLASHB_ADDR10	U28.F41	U16.6
FLASHB_ADDR11	U28.F42	U16.5
FLASHB_ADDR12	U28.E41	U16.4
FLASHB_ADDR13	U28.E42	U16.3
FLASHB_ADDR14	U28.D41	U16.2
FLASHB_ADDR15	U28.D42	U16.1
FLASHB_ADDR16	U28.AA33	U16.48
FLASHB_ADDR17	U28.M41	U16.17
FLASHB_ADDR18	U28.L41	U16.16
FLASHB_ADDR19	U28.L42	U16.15
FLASHB_ADDR20	U28.J42	U16.10
FLASHB_ADDR21	U28.H41	U16.9
FLASHB_DATA0	U28.V41	U16.29
FLASHB_DATA1	U28.W41	U16.31
FLASHB_DATA2	U28.Y39	U16.33
FLASHB_DATA3	U28.Y36	U16.35
FLASHB_DATA4	U28.Y33	U16.38
FLASHB_DATA5	U28.Y31	U16.40
FLASHB_DATA6	U28.AA39	U16.42
FLASHB_DATA7	U28.AA36	U16.44
FLASHB_DATA8	U28.W42	U16.30

Signal Name	FPGA Pin	FLASH
FLASHB_DATA9	U28.Y40	U16.32
FLASHB_DATA10	U28.Y37	U16.34
FLASHB_DATA11	U28.Y34	U16.36
FLASHB_DATA12	U28.Y32	U16.39
FLASHB_DATA13	U28.AA40	U16.41
FLASHB_DATA14	U28.AA37	U16.43
FLASHB_DATA15	U28.AA34	U16.45
FLASHB_CEN	U28.U41	U16.26
FLASHB_OEN	U28.V42	U16.28
FLASHB_WEN	U28.J41	U16.11
FLASHB_WPN	U28.K41	U16.14
FLASHC_ADDR0	U29.AP39	U17.25
FLASHC_ADDR1	U29.AN38	U17.24
FLASHC_ADDR2	U29.AN40	U17.23
FLASHC_ADDR3	U29.AM38	U17.22
FLASHC_ADDR4	U29.AM39	U17.21
FLASHC_ADDR5	U29.AM40	U17.20
FLASHC_ADDR6	U29.AL38	U17.19
FLASHC_ADDR7	U29.AL39	U17.18
FLASHC_ADDR8	U29.AG38	U17.8
FLASHC_ADDR9	U29.AG39	U17.7
FLASHC_ADDR10	U29.AG40	U17.6
FLASHC_ADDR11	U29.AF39	U17.5
FLASHC_ADDR12	U29.AF40	U17.4
FLASHC_ADDR13	U29.AE39	U17.3
FLASHC_ADDR14	U29.AD39	U17.2
FLASHC_ADDR15	U29.AD40	U17.1
FLASHC_ADDR16	U29.AW40	U17.48
FLASHC_ADDR17	U29.AL40	U17.17

Signal Name	FPGA Pin	FLASH
FLASHC_ADDR18	U29.AK38	U17.16
FLASHC_ADDR19	U29.AK39	U17.15
FLASHC_ADDR20	U29.AH38	U17.10
FLASHC_ADDR21	U29.AH40	U17.9
FLASHC_DATA0	U29.AR39	U17.29
FLASHC_DATA1	U29.AT42	U17.31
FLASHC_DATA2	U29.AT40	U17.33
FLASHC_DATA3	U29.AT38	U17.35
FLASHC_DATA4	U29.AU41	U17.38
FLASHC_DATA5	U29.AU39	U17.40
FLASHC_DATA6	U29.AV41	U17.42
FLASHC_DATA7	U29.AW42	U17.44
FLASHC_DATA8	U29.AR38	U17.30
FLASHC_DATA9	U29.AT41	U17.32
FLASHC_DATA10	U29.AT39	U17.34
FLASHC_DATA11	U29.AU42	U17.36
FLASHC_DATA12	U29.AU40	U17.39
FLASHC_DATA13	U29.AV42	U17.41
FLASHC_DATA14	U29.AV40	U17.43
FLASHC_DATA15	U29.AW41	U17.45
FLASHC_CEN	U29.AP38	U17.26
FLASHC_OEN	U29.AR40	U17.28
FLASHC_WEN	U29.AJ40	U17.11
FLASHC_WPN	U29.AK40	U17.14
FLASHG_ADDR0	U80.M3	U88.25
FLASHG_ADDR1	U80.M2	U88.24
FLASHG_ADDR2	U80.L3	U88.23
FLASHG_ADDR3	U80.L2	U88.22
FLASHG_ADDR4	U80.L1	U88.21

Signal Name	FPGA Pin	FLASH
FLASHG_ADDR5	U80.K3	U88.20
FLASHG_ADDR6	U80.K2	U88.19
FLASHG_ADDR7	U80.K1	U88.18
FLASHG_ADDR8	U80.F2	U88.8
FLASHG_ADDR9	U80.F1	U88.7
FLASHG_ADDR10	U80.E3	U88.6
FLASHG_ADDR11	U80.E2	U88.5
FLASHG_ADDR12	U80.E1	U88.4
FLASHG_ADDR13	U80.D3	U88.3
FLASHG_ADDR14	U80.D2	U88.2
FLASHG_ADDR15	U80.D1	U88.1
FLASHG_ADDR16	U80.W3	U88.48
FLASHG_ADDR17	U80.J2	U88.17
FLASHG_ADDR18	U80.J1	U88.16
FLASHG_ADDR19	U80.H3	U88.15
FLASHG_ADDR20	U80.G1	U88.10
FLASHG_ADDR21	U80.F3	U88.9
FLASHG_DATA0	U80.N3	U88.29
FLASHG_DATA1	U80.P2	U88.31
FLASHG_DATA2	U80.R1	U88.33
FLASHG_DATA3	U80.R3	U88.35
FLASHG_DATA4	U80.T3	U88.38
FLASHG_DATA5	U80.U2	U88.40
FLASHG_DATA6	U80.V1	U88.42
FLASHG_DATA7	U80.W1	U88.44
FLASHG_DATA8	U80.P1	U88.30
FLASHG_DATA9	U80.P3	U88.32
FLASHG_DATA10	U80.R2	U88.34
FLASHG_DATA11	U80.T2	U88.36

Signal Name	FPGA Pin	FLASH
FLASHG_DATA12	U80.U1	U88.39
FLASHG_DATA13	U80.U3	U88.41
FLASHG_DATA14	U80.V2	U88.43
FLASHG_DATA15	U80.W2	U88.45
FLASHG_CEN	U80.N1	U88.26
FLASHG_OEN	U80.N2	U88.28
FLASHG_WEN	U80.G2	U88.11
FLASHG_WPN	U80.H2	U88.14
FLASHI_ADDR0	U79.AM3	U86.25
FLASHI_ADDR1	U79.AM2	U86.24
FLASHI_ADDR2	U79.AM1	U86.23
FLASHI_ADDR3	U79.AL3	U86.22
FLASHI_ADDR4	U79.AL2	U86.21
FLASHI_ADDR5	U79.AK3	U86.20
FLASHI_ADDR6	U79.AK2	U86.19
FLASHI_ADDR7	U79.AK1	U86.18
FLASHI_ADDR8	U79.AF3	U86.8
FLASHI_ADDR9	U79.AF2	U86.7
FLASHI_ADDR10	U79.AF1	U86.6
FLASHI_ADDR11	U79.AE2	U86.5
FLASHI_ADDR12	U79.AE1	U86.4
FLASHI_ADDR13	U79.AD3	U86.3
FLASHI_ADDR14	U79.AD2	U86.2
FLASHI_ADDR15	U79.AD1	U86.1
FLASHI_ADDR16	U79.AW3	U86.48
FLASHI_ADDR17	U79.AJ3	U86.17
FLASHI_ADDR18	U79.AJ2	U86.16
FLASHI_ADDR19	U79.AJ1	U86.15
FLASHI_ADDR20	U79.AG3	U86.10

Signal Name	FPGA Pin	FLASH
FLASHI_ADDR21	U79.AG2	U86.9
FLASHI_DATA0	U79.AN3	U86.29
FLASHI_DATA1	U79.AP2	U86.31
FLASHI_DATA2	U79.AR3	U86.33
FLASHI_DATA3	U79.AT2	U86.35
FLASHI_DATA4	U79.AU1	U86.38
FLASHI_DATA5	U79.AU3	U86.40
FLASHI_DATA6	U79.AV2	U86.42
FLASHI_DATA7	U79.AW1	U86.44
FLASHI_DATA8	U79.AP1	U86.30
FLASHI_DATA9	U79.AR2	U86.32
FLASHI_DATA10	U79.AT1	U86.34
FLASHI_DATA11	U79.AT3	U86.36
FLASHI_DATA12	U79.AU2	U86.39
FLASHI_DATA13	U79.AV1	U86.41
FLASHI_DATA14	U79.AV3	U86.43
FLASHI_DATA15	U79.AW2	U86.45
FLASHI_CEN	U79.AN1	U86.26
FLASHI_OEN	U79.AN2	U86.28
FLASHI_WEN	U79.AH1	U86.11
FLASHI_WPN	U79.AH3	U86.14

6.2 DDR SDRAM

Double Data Rate (DDR) SDRAM represents an enhancement to the traditional SDRAM. Instead of data and control signals operating at the same frequency, data operates at twice the clock frequency, while address and control operate at the base clock frequency. In other words, the data is written or read from the device on every clock transition, or twice per clock cycle. This effectively doubles the throughput of the memory device.

The trade-off for such an improvement in throughput is increased complexity in interface logic to the DDR memory, as well as increased complexity in routing the

DDR signals on the printed circuit board. Additionally, this memory has the same latencies as standard SDRAM, so that while the data transfers are twice as fast, the latencies associated with DDR SDRAM are on par with standard SDRAM.

6.2.1 Basics of DDR Operation

DDR SDRAM provides data capture at a rate of twice the clock frequency. Therefore, DDR SDRAM with a clock frequency of 100 MHz has a peak data transfer rate of 200 MHz or 6.4 Gigabits per second for a 16-bit interface. In order to maintain high-speed signal integrity and stringent timing goals, a bi-directional data strobe is used in conjunction with SSTL_2 signaling standard as well as differential clocks. DDR SDRAM operates as a source-synchronous system, in which data is captured twice per clock cycle, using a bi-directional data strobe to clock the data. The DDR SDRAM control bus consists of a clock enable, chip select, row and column addresses, bank address, and a write enable. Commands are entered on the positive edges of the clock, and data occurs for both positive and negative edges of the clock. The double data rate memory utilizes a differential pair for the system clock and, therefore, has both a true clock (CKp) and complementary clock (CKn) signal.

6.2.2 DDR SDRAM Configuration

The DDR SDRAM memory components on the DN6000K10 are arranged as a 16-bit mode, refer to Figure 32 as an example of a DDR interface (shown is the DRR device on FPGA A). Each FPGA has two discrete parts (U22, U32 etc). The components used are 64Mb x 16 parts, organized as 16 million deep by 16-bits wide and 4 banks (for more information, refer to Micron's datasheet PN MT46V64M16).

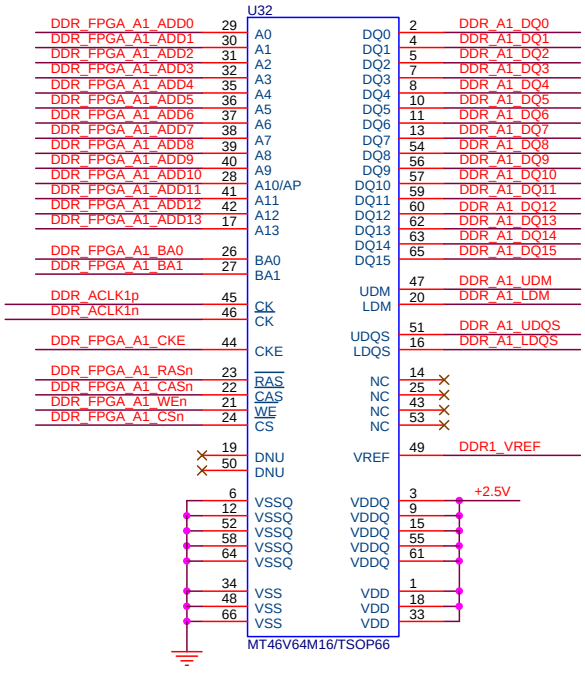


Figure 32 - DDR SDRAM Connection

6.2.3 DDR SDRAM Clocking

Refer to the DDR Clocking Section.

6.2.4 DDR SDRAM Termination

DDR SDRAM is based on the SSTL2 (JEDEC Standard - Stub Series Terminated Logic for 2.5V) signaling standard. The SSTL2 termination model used for DDR SDRAM has two types of termination:

- Class 1
  - Also called SSTL2_I
  - Used for unidirectional signaling (Control signals)
- Class 2
  - Also called SSTL2_II
  - Used for bi-directional signaling (Data signals)

Both Class 1 and Class 2 are based on a 50Ω controlled impedance environment, and termination to VTT, a 1.25V power supply.

SSTL2 Class 1 termination is used for unidirectional signaling, such as control signals. It is based on a 50Ω controlled impedance driver, a 50Ω controlled impedance transmission line, and a 50Ω parallel termination to VTT at the receiver. Figure 33 shows a basic SSTL2 Class 1 circuit. The driver is brought to 50Ω by the addition of a 25Ω series resistor immediately adjacent to the driver (implemented using DCI, thus no need for an external component).

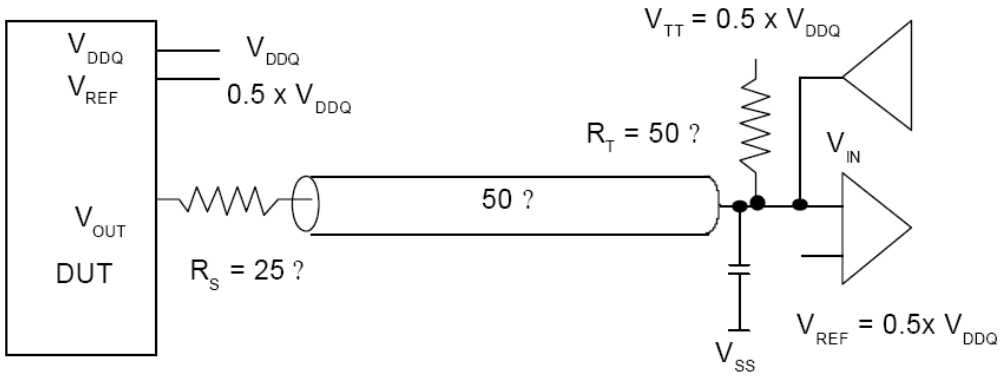


Figure 33 - SSTL2 Class 1 Termination

SSTL2 Class 2 termination is used for bi-directional signaling, such as data signals. It is based on a 50Ω controlled impedance driver and a 50Ω parallel termination to VTT for the receiver at both ends, connected through a 50Ω controlled impedance

transmission line. Figure 34 shows a basic SSTL2 Class 2 circuit. The driver is brought to 50Ω by the addition of a 25Ω series resistor immediately adjacent to the driver.

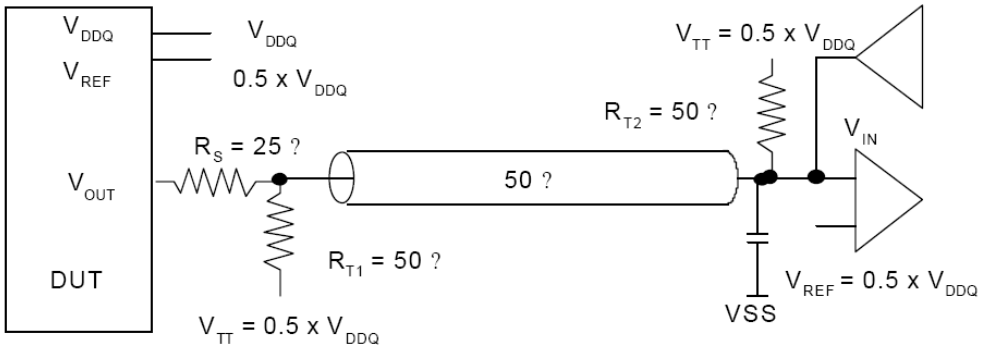


Figure 34 - SSTL2 Class 2 Termination

Note: DCI termination must be implemented in the DDR SDRAM controller design.

6.2.5 DDR SDRAM Power Supply

The DATEL +2.5V module (U14, U64, U89) is used to supply power to the +2.5V plane that supplies the VDDQ pins of the DDR SDRAM devices. Due to the power requirements, three separate PSU's are used to supply the power to the DRR devices on FPGA A/C, FPGA D/F and FPGA G/I. According to the JEDEC Specification – Double Data Rate (DDR) SDRAM termination voltage VTT must track 50% of VDDQ over voltage, temperature and noise. The ML6554 (U14) is used as a voltage source for DDR termination. Connecting the V_{REF} pin to the +2.5V supply allows the regulator to track the VDDQ supply (refer to Figure 35). A dedicated VREF output supplies the VREF pins on the FPGA as well as on the DDR SDRAM devices and maintains a less that 40mV offset from VTT.

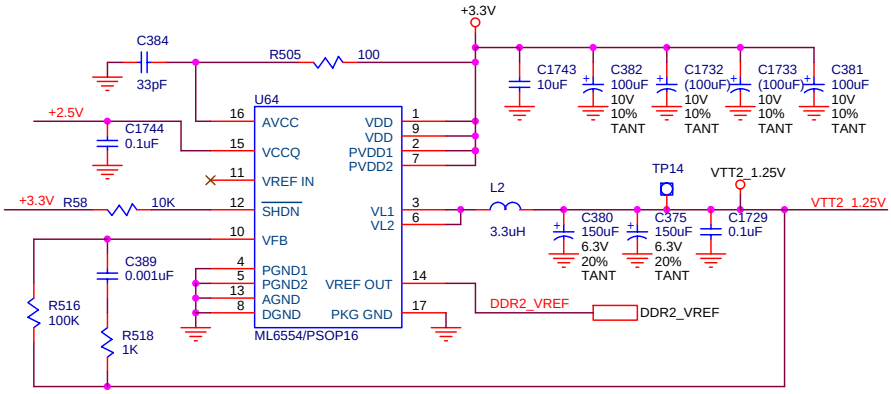


Figure 35 - DDR VTT Termination Regulator

6.2.6 DDR SDRAM Connection to the FPGA

The connections between the FPGA and the DDR SDRAM are not homogeneous, as control and address are handled differently from the data and differently from the clocks. However, all of these signals are controlled impedance, and are SSTL2 terminated. The termination of these signals is covered in DDR SDRAM Termination.

The Data signals (DQ), the Data Strobe (DQS) and the Data Mask (DM) signals are point-to-point signals, going from the FPGA to the DDR SDRAM components. As mentioned above, these signals are controlled impedance, and terminated according to the DDR SDRAM specification. The data, data strobe, and data mask signals all serve different purposes. The data signals are self-evident, carrying the raw data between the chips, and are bi-directional. The data strobe signals are responsible for actual clocking in the data on rising and falling edges of the clock. Finally, the data mask signals can be used to enable or disable the reading and writing of some of the bytes in a 16-bit word transaction. The interface signals between the FPGA and the DDR SDRAM components in covered in Table 18.

Table 18 - Connection between FPGA's and DDR SDRAM's

Signal Name	FPGA Pin	DDR SDRAM
DDR_A1_DATA0	U27.H24	U32.2
DDR_A1_DATA1	U27.G24	U32.4
DDR_A1_DATA2	U27.K24	U32.5
DDR_A1_DATA3	U27.J24	U32.7
DDR_A1_DATA4	U27.M24	U32.8
DDR_A1_DATA5	U27.L24	U32.10
DDR_A1_DATA6	U27.H25	U32.11
DDR_A1_DATA7	U27.G25	U32.13

Signal Name	FPGA Pin	DDR SDRAM
DDR_A1_DATA8	U27.G26	U32.54
DDR_A1_DATA9	U27.F26	U32.56
DDR_A1_DATA10	U27.J26	U32.57
DDR_A1_DATA11	U27.H26	U32.59
DDR_A1_DATA12	U27.K26	U32.60
DDR_A1_DATA13	U27.L26	U32.62
DDR_A1_DATA14	U27.M26	U32.63
DDR_A1_DATA15	U27.M25	U32.65
DDR_FPGA_A1_ADD0	U27.F24	U32.29
DDR_FPGA_A1_ADD1	U27.E24	U32.30
DDR_FPGA_A1_ADD2	U27.L25	U32.31
DDR_FPGA_A1_ADD3	U27.K25	U32.32
DDR_FPGA_A1_ADD4	U27.G27	U32.35
DDR_FPGA_A1_ADD5	U27.H27	U32.36
DDR_FPGA_A1_ADD6	U27.K27	U32.37
DDR_FPGA_A1_ADD7	U27.J27	U32.38
DDR_FPGA_A1_ADD8	U27.M27	U32.39
DDR_FPGA_A1_ADD9	U27.L27	U32.40
DDR_FPGA_A1_ADD10	U27.C28	U32.28
DDR_FPGA_A1_ADD11	U27.C29	U32.41
DDR_FPGA_A1_ADD12	U27.F28	U32.42
DDR_FPGA_A1_ADD13	U27.E28	U32.17
DDR_FPGA_A1_UDQS	U27.E27	U32.51
DDR_FPGA_A1_LDQS	U27.E26	U32.16
DDR_FPGA_A1_UDM	U27.D27	U32.47
DDR_FPGA_A1_LDM	U27.C25	U32.20
DDR_FPGA_A1_BA0	U27.H23	U32.26
DDR_FPGA_A1_BA1	U27.J23	U32.27
DDR_FPGA_A1_CASN	U27.C24	U32.22

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_A1_CKE	U27.D26	U32.44
DDR_FPGA_A1_CSN	U27.L23	U32.24
DDR_FPGA_A1_RASN	U27.D24	U32.23
DDR_FPGA_A1_WEN	U27.K23	U32.21
DDR_A2_DATA0	U27.H30	U22.2
DDR_A2_DATA1	U27.G30	U22.4
DDR_A2_DATA2	U27.K30	U22.5
DDR_A2_DATA3	U27.J30	U22.7
DDR_A2_DATA4	U27.M30	U22.8
DDR_A2_DATA5	U27.L30	U22.10
DDR_A2_DATA6	U27.J31	U22.11
DDR_A2_DATA7	U27.H31	U22.13
DDR_A2_DATA8	U27.F32	U22.54
DDR_A2_DATA9	U27.E32	U22.56
DDR_A2_DATA10	U27.D33	U22.57
DDR_A2_DATA11	U27.E33	U22.59
DDR_A2_DATA12	U27.G33	U22.60
DDR_A2_DATA13	U27.F33	U22.62
DDR_A2_DATA14	U27.J33	U22.63
DDR_A2_DATA15	U27.H33	U22.65
DDR_FPGA_A2_ADD0	U27.H28	U22.29
DDR_FPGA_A2_ADD1	U27.K28	U22.30
DDR_FPGA_A2_ADD2	U27.L28	U22.31
DDR_FPGA_A2_ADD3	U27.E29	U22.32
DDR_FPGA_A2_ADD4	U27.D29	U22.35
DDR_FPGA_A2_ADD5	U27.G29	U22.36
DDR_FPGA_A2_ADD6	U27.F29	U22.37
DDR_FPGA_A2_ADD7	U27.H29	U22.38
DDR_FPGA_A2_ADD8	U27.L29	U22.39

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_A2_ADD9	U27.K29	U22.40
DDR_FPGA_A2_ADD10	U27.C30	U22.28
DDR_FPGA_A2_ADD11	U27.D30	U22.41
DDR_FPGA_A2_ADD12	U27.F30	U22.42
DDR_FPGA_A2_ADD13	U27.E30	U22.17
DDR_FPGA_A2_UDQS	U27.C34	U22.51
DDR_FPGA_A2_LDQS	U27.G31	U22.16
DDR_FPGA_A2_UDM	U27.K32	U22.20
DDR_FPGA_A2_LDM	U27.D31	U22.47
DDR_FPGA_A2_BA0	U27.E34	U22.26
DDR_FPGA_A2_BA1	U27.F34	U22.27
DDR_FPGA_A2_CASN	U27.C32	U22.22
DDR_FPGA_A2_CKE	U27.H32	U22.44
DDR_FPGA_A2_CSN	U27.K31	U22.24
DDR_FPGA_A2_RASN	U27.C33	U22.23
DDR_FPGA_A2_WEN	U27.L31	U22.21
DDR_C1_DATA0	U29.AW33	U23.2
DDR_C1_DATA1	U29.AV33	U23.4
DDR_C1_DATA2	U29.AY32	U23.5
DDR_C1_DATA3	U29.AY33	U23.7
DDR_C1_DATA4	U29.AU32	U23.8
DDR_C1_DATA5	U29.AV32	U23.10
DDR_C1_DATA6	U29.AM31	U23.11
DDR_C1_DATA7	U29.AN31	U23.13
DDR_C1_DATA8	U29.AU31	U23.54
DDR_C1_DATA9	U29.AT31	U23.56
DDR_C1_DATA10	U29.AN30	U23.57
DDR_C1_DATA11	U29.AP30	U23.59
DDR_C1_DATA12	U29.AL30	U23.60

Signal Name	FPGA Pin	DDR SDRAM
DDR_C1_DATA13	U29.AM30	U23.62
DDR_C1_DATA14	U29.AR30	U23.63
DDR_C1_DATA15	U29.AT30	U23.65
DDR_FPGA_C1_ADD0	U29.AY30	U23.29
DDR_FPGA_C1_ADD1	U29.AW30	U23.30
DDR_FPGA_C1_ADD2	U29.AU30	U23.31
DDR_FPGA_C1_ADD3	U29.AV30	U23.32
DDR_FPGA_C1_ADD4	U29.AU28	U23.35
DDR_FPGA_C1_ADD5	U29.AV28	U23.36
DDR_FPGA_C1_ADD6	U29.AL27	U23.37
DDR_FPGA_C1_ADD7	U29.AM27	U23.38
DDR_FPGA_C1_ADD8	U29.AT27	U23.39
DDR_FPGA_C1_ADD9	U29.AR27	U23.40
DDR_FPGA_C1_ADD10	U29.AN27	U23.28
DDR_FPGA_C1_ADD11	U29.AP27	U23.41
DDR_FPGA_C1_ADD12	U29.AN26	U23.42
DDR_FPGA_C1_ADD13	U29.AM26	U23.17
DDR_FPGA_C1_UDQS	U29.AP31	U23.51
DDR_FPGA_C1_LDQS	U29.AN32	U23.16
DDR_FPGA_C1_UDM	U29.AW31	U23.47
DDR_FPGA_C1_LDM	U29.AU33	U23.20
DDR_FPGA_C1_BA0	U29.AM23	U23.26
DDR_FPGA_C1_BA1	U29.AN23	U23.27
DDR_FPGA_C1_CASN	U29.AP23	U23.22
DDR_FPGA_C1_CKE	U29.AR32	U23.44
DDR_FPGA_C1_CSN	U29.AL28	U23.24
DDR_FPGA_C1_RASN	U29.AR23	U23.23
DDR_FPGA_C1_WEN	U29.AV27	U23.21
DDR_C2_DATA0	U29.AT29	U34.2

Signal Name	FPGA Pin	DDR SDRAM
DDR_C2_DATA1	U29.AU29	U34.4
DDR_C2_DATA2	U29.AN28	U34.5
DDR_C2_DATA3	U29.AM28	U34.7
DDR_C2_DATA4	U29.AV29	U34.8
DDR_C2_DATA5	U29.AW29	U34.10
DDR_C2_DATA6	U29.AY28	U34.11
DDR_C2_DATA7	U29.AY29	U34.13
DDR_C2_DATA8	U29.AM25	U34.54
DDR_C2_DATA9	U29.AN25	U34.56
DDR_C2_DATA10	U29.AV25	U34.57
DDR_C2_DATA11	U29.AV26	U34.59
DDR_C2_DATA12	U29.AR25	U34.60
DDR_C2_DATA13	U29.AT25	U34.62
DDR_C2_DATA14	U29.AN24	U34.63
DDR_C2_DATA15	U29.AP24	U34.65
DDR_FPGA_C2_ADD0	U29.AL26	U34.29
DDR_FPGA_C2_ADD1	U29.AL25	U34.30
DDR_FPGA_C2_ADD2	U29.AP26	U34.31
DDR_FPGA_C2_ADD3	U29.AR26	U34.32
DDR_FPGA_C2_ADD4	U29.AT26	U34.35
DDR_FPGA_C2_ADD5	U29.AU26	U34.36
DDR_FPGA_C2_ADD6	U29.AL24	U34.37
DDR_FPGA_C2_ADD7	U29.AM24	U34.38
DDR_FPGA_C2_ADD8	U29.AR24	U34.39
DDR_FPGA_C2_ADD9	U29.AT24	U34.40
DDR_FPGA_C2_ADD10	U29.AY24	U34.28
DDR_FPGA_C2_ADD11	U29.AW24	U34.41
DDR_FPGA_C2_ADD12	U29.AU24	U34.42
DDR_FPGA_C2_ADD13	U29.AV24	U34.17

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_C2_UDQS	U29.AW26	U34.51
DDR_FPGA_C2_LDQS	U29.AR29	U34.16
DDR_FPGA_C2_UDM	U29.AY25	U34.47
DDR_FPGA_C2_LDM	U29.AR28	U34.20
DDR_FPGA_C2_BA0	U29.AN29	U34.26
DDR_FPGA_C2_BA1	U29.AM29	U34.27
DDR_FPGA_C2_CASN	U29.AY23	U34.22
DDR_FPGA_C2_CKE	U29.AW27	U34.44
DDR_FPGA_C2_CSN	U29.AL22	U34.24
DDR_FPGA_C2_RASN	U29.AW23	U34.23
DDR_FPGA_C2_WEN	U29.AV23	U34.21
DDR_D1_DATA0	U53.AN20	U46.2
DDR_D1_DATA1	U53.AM20	U46.4
DDR_D1_DATA2	U53.AP20	U46.5
DDR_D1_DATA3	U53.AR20	U46.7
DDR_D1_DATA4	U53.AV19	U46.8
DDR_D1_DATA5	U53.AU19	U46.10
DDR_D1_DATA6	U53.AW19	U46.11
DDR_D1_DATA7	U53.AY19	U46.13
DDR_D1_DATA8	U53.AT18	U46.54
DDR_D1_DATA9	U53.AR18	U46.56
DDR_D1_DATA10	U53.AV17	U46.57
DDR_D1_DATA11	U53.AV18	U46.59
DDR_D1_DATA12	U53.AN18	U46.60
DDR_D1_DATA13	U53.AM18	U46.62
DDR_D1_DATA14	U53.AU17	U46.63
DDR_D1_DATA15	U53.AT17	U46.65
DDR_FPGA_D1_ADD0	U53.AT19	U46.29
DDR_FPGA_D1_ADD1	U53.AR19	U46.30

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_D1_ADD2	U53.AM19	U46.31
DDR_FPGA_D1_ADD3	U53.AL19	U46.32
DDR_FPGA_D1_ADD4	U53.AP19	U46.35
DDR_FPGA_D1_ADD5	U53.AN19	U46.36
DDR_FPGA_D1_ADD6	U53.AR17	U46.37
DDR_FPGA_D1_ADD7	U53.AP17	U46.38
DDR_FPGA_D1_ADD8	U53.AL18	U46.39
DDR_FPGA_D1_ADD9	U53.AL17	U46.40
DDR_FPGA_D1_ADD10	U53.AM17	U46.28
DDR_FPGA_D1_ADD11	U53.AN17	U46.41
DDR_FPGA_D1_ADD12	U53.AP16	U46.42
DDR_FPGA_D1_ADD13	U53.AN16	U46.17
DDR_FPGA_D1_UDQS	U53.AY18	U46.51
DDR_FPGA_D1_LDQS	U53.AV20	U46.16
DDR_FPGA_D1_UDM	U53.AW17	U46.47
DDR_FPGA_D1_LDM	U53.AL21	U46.20
DDR_FPGA_D1_BA0	U53.AW20	U46.26
DDR_FPGA_D1_BA1	U53.AY20	U46.27
DDR_FPGA_D1_CASN	U53.AT16	U46.22
DDR_FPGA_D1_CKE	U53.AW16	U46.44
DDR_FPGA_D1_CSN	U53.AV16	U46.24
DDR_FPGA_D1_RASN	U53.AR16	U46.23
DDR_FPGA_D1_WEN	U53.AL15	U46.21
DDR_D2_DATA0	U53.AW14	U58.2
DDR_D2_DATA1	U53.AV14	U58.4
DDR_D2_DATA2	U53.AM15	U58.5
DDR_D2_DATA3	U53.AN15	U58.7
DDR_D2_DATA4	U53.AU14	U58.8
DDR_D2_DATA5	U53.AT14	U58.10

Signal Name	FPGA Pin	DDR SDRAM
DDR_D2_DATA6	U53.AN14	U58.11
DDR_D2_DATA7	U53.AM14	U58.13
DDR_D2_DATA8	U53.AM13	U58.54
DDR_D2_DATA9	U53.AL13	U58.56
DDR_D2_DATA10	U53.AP13	U58.57
DDR_D2_DATA11	U53.AN13	U58.59
DDR_D2_DATA12	U53.AR12	U58.60
DDR_D2_DATA13	U53.AP12	U58.62
DDR_D2_DATA14	U53.AT12	U58.63
DDR_D2_DATA15	U53.AU12	U58.65
DDR_FPGA_D2_ADD0	U53.AV15	U58.29
DDR_FPGA_D2_ADD1	U53.AU15	U58.30
DDR_FPGA_D2_ADD2	U53.AY14	U58.31
DDR_FPGA_D2_ADD3	U53.AY15	U58.32
DDR_FPGA_D2_ADD4	U53.AV13	U58.35
DDR_FPGA_D2_ADD5	U53.AU13	U58.36
DDR_FPGA_D2_ADD6	U53.AW13	U58.37
DDR_FPGA_D2_ADD7	U53.AY13	U58.38
DDR_FPGA_D2_ADD8	U53.AN12	U58.39
DDR_FPGA_D2_ADD9	U53.AM12	U58.40
DDR_FPGA_D2_ADD10	U53.AV11	U58.28
DDR_FPGA_D2_ADD11	U53.AU11	U58.41
DDR_FPGA_D2_ADD12	U53.AY10	U58.42
DDR_FPGA_D2_ADD13	U53.AY11	U58.17
DDR_FPGA_D2_UDQS	U53.AT13	U58.51
DDR_FPGA_D2_LDQS	U53.AR15	U58.16
DDR_FPGA_D2_UDM	U53.AW12	U58.47
DDR_FPGA_D2_LDM	U53.AR14	U58.20
DDR_FPGA_D2_BA0	U53.AL16	U58.26

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_D2_BA1	U53.AM16	U58.27
DDR_FPGA_D2_CASN	U53.AW10	U58.22
DDR_FPGA_D2_CKE	U53.AN11	U58.44
DDR_FPGA_D2_CSN	U53.AR11	U58.24
DDR_FPGA_D2_RASN	U53.AV10	U58.23
DDR_FPGA_D2_WEN	U53.AU10	U58.21
DDR_F1_DATA0	U51.M18	U45.2
DDR_F1_DATA1	U51.M17	U45.4
DDR_F1_DATA2	U51.L17	U45.5
DDR_F1_DATA3	U51.K17	U45.7
DDR_F1_DATA4	U51.H17	U45.8
DDR_F1_DATA5	U51.J17	U45.10
DDR_F1_DATA6	U51.F17	U45.11
DDR_F1_DATA7	U51.G17	U45.13
DDR_F1_DATA8	U51.K18	U45.54
DDR_F1_DATA9	U51.L18	U45.56
DDR_F1_DATA10	U51.G18	U45.57
DDR_F1_DATA11	U51.H18	U45.59
DDR_F1_DATA12	U51.E17	U45.60
DDR_F1_DATA13	U51.E18	U45.62
DDR_F1_DATA14	U51.J19	U45.63
DDR_F1_DATA15	U51.K19	U45.65
DDR_FPGA_F1_ADD0	U51.C14	U45.29
DDR_FPGA_F1_ADD1	U51.C15	U45.30
DDR_FPGA_F1_ADD2	U51.L16	U45.31
DDR_FPGA_F1_ADD3	U51.M16	U45.32
DDR_FPGA_F1_ADD4	U51.J16	U45.35
DDR_FPGA_F1_ADD5	U51.K16	U45.36
DDR_FPGA_F1_ADD6	U51.H16	U45.37

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_F1_ADD7	U51.G16	U45.38
DDR_FPGA_F1_ADD8	U51.G19	U45.39
DDR_FPGA_F1_ADD9	U51.H19	U45.40
DDR_FPGA_F1_ADD10	U51.E19	U45.28
DDR_FPGA_F1_ADD11	U51.F19	U45.41
DDR_FPGA_F1_ADD12	U51.D19	U45.42
DDR_FPGA_F1_ADD13	U51.C19	U45.17
DDR_FPGA_F1_UDQS	U51.M19	U45.51
DDR_FPGA_F1_LDQS	U51.D17	U45.16
DDR_FPGA_F1_UDM	U51.C18	U45.47
DDR_FPGA_F1_LDM	U51.D16	U45.20
DDR_FPGA_F1_BA0	U51.C20	U45.26
DDR_FPGA_F1_BA1	U51.D20	U45.27
DDR_FPGA_F1_CASN	U51.K20	U45.22
DDR_FPGA_F1_CKE	U51.M21	U45.44
DDR_FPGA_F1_CSN	U51.J20	U45.24
DDR_FPGA_F1_RASN	U51.L20	U45.23
DDR_FPGA_F1_WEN	U51.H20	U45.21
DDR_F2_DATA0	U51.H10	U57.2
DDR_F2_DATA1	U51.J10	U57.4
DDR_F2_DATA2	U51.F10	U57.5
DDR_F2_DATA3	U51.G10	U57.7
DDR_F2_DATA4	U51.E10	U57.8
DDR_F2_DATA5	U51.D10	U57.10
DDR_F2_DATA6	U51.C10	U57.11
DDR_F2_DATA7	U51.C11	U57.13
DDR_F2_DATA8	U51.K14	U57.54
DDR_F2_DATA9	U51.L14	U57.56
DDR_F2_DATA10	U51.F14	U57.57

Signal Name	FPGA Pin	DDR SDRAM
DDR_F2_DATA11	U51.G14	U57.59
DDR_F2_DATA12	U51.D14	U57.60
DDR_F2_DATA13	U51.E14	U57.62
DDR_F2_DATA14	U51.L15	U57.63
DDR_F2_DATA15	U51.K15	U57.65
DDR_FPGA_F2_ADD0	U51.G12	U57.29
DDR_FPGA_F2_ADD1	U51.F12	U57.30
DDR_FPGA_F2_ADD2	U51.D12	U57.31
DDR_FPGA_F2_ADD3	U51.L13	U57.32
DDR_FPGA_F2_ADD4	U51.M13	U57.35
DDR_FPGA_F2_ADD5	U51.J13	U57.36
DDR_FPGA_F2_ADD6	U51.K13	U57.37
DDR_FPGA_F2_ADD7	U51.G13	U57.38
DDR_FPGA_F2_ADD8	U51.H13	U57.39
DDR_FPGA_F2_ADD9	U51.E13	U57.40
DDR_FPGA_F2_ADD10	U51.F13	U57.28
DDR_FPGA_F2_ADD11	U51.D13	U57.41
DDR_FPGA_F2_ADD12	U51.C13	U57.42
DDR_FPGA_F2_ADD13	U51.M15	U57.17
DDR_FPGA_F2_UDQS	U51.F15	U57.51
DDR_FPGA_F2_LDQS	U51.F11	U57.16
DDR_FPGA_F2_UDM	U51.H15	U57.47
DDR_FPGA_F2_LDM	U51.H11	U57.20
DDR_FPGA_F2_BA0	U51.F9	U57.26
DDR_FPGA_F2_BA1	U51.E9	U57.27
DDR_FPGA_F2_CASN	U51.H12	U57.22
DDR_FPGA_F2_CKE	U51.K11	U57.44
DDR_FPGA_F2_CSN	U51.K12	U57.24
DDR_FPGA_F2_RASN	U51.J12	U57.23

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_F2_WEN	U51.L12	U57.21
DDR_G1_DATA0	U80.M18	U76.2
DDR_G1_DATA1	U80.M17	U76.4
DDR_G1_DATA2	U80.L17	U76.5
DDR_G1_DATA3	U80.K17	U76.7
DDR_G1_DATA4	U80.H17	U76.8
DDR_G1_DATA5	U80.J17	U76.10
DDR_G1_DATA6	U80.F17	U76.11
DDR_G1_DATA7	U80.G17	U76.13
DDR_G1_DATA8	U80.K18	U76.54
DDR_G1_DATA9	U80.L18	U76.56
DDR_G1_DATA10	U80.G18	U76.57
DDR_G1_DATA11	U80.H18	U76.59
DDR_G1_DATA12	U80.E17	U76.60
DDR_G1_DATA13	U80.E18	U76.62
DDR_G1_DATA14	U80.J19	U76.63
DDR_G1_DATA15	U80.K19	U76.65
DDR_FPGA_G1_ADD0	U80.C14	U76.29
DDR_FPGA_G1_ADD1	U80.C15	U76.30
DDR_FPGA_G1_ADD2	U80.L16	U76.31
DDR_FPGA_G1_ADD3	U80.M16	U76.32
DDR_FPGA_G1_ADD4	U80.J16	U76.35
DDR_FPGA_G1_ADD5	U80.K16	U76.36
DDR_FPGA_G1_ADD6	U80.H16	U76.37
DDR_FPGA_G1_ADD7	U80.G16	U76.38
DDR_FPGA_G1_ADD8	U80.G19	U76.39
DDR_FPGA_G1_ADD9	U80.H19	U76.40
DDR_FPGA_G1_ADD10	U80.E19	U76.28
DDR_FPGA_G1_ADD11	U80.F19	U76.41

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_G1_ADD12	U80.D19	U76.42
DDR_FPGA_G1_ADD13	U80.C19	U76.17
DDR_FPGA_G1_UDQS	U80.M19	U76.51
DDR_FPGA_G1_LDQS	U80.D17	U76.16
DDR_FPGA_G1_UDM	U80.C18	U76.47
DDR_FPGA_G1_LDM	U80.D16	U76.20
DDR_FPGA_G1_BA0	U80.C20	U76.26
DDR_FPGA_G1_BA1	U80.D20	U76.27
DDR_FPGA_G1_CASN	U80.K20	U76.22
DDR_FPGA_G1_CKE	U80.M21	U76.44
DDR_FPGA_G1_CSN	U80.J20	U76.24
DDR_FPGA_G1_RASN	U80.L20	U76.23
DDR_FPGA_G1_WEN	U80.H20	U76.21
DDR_G2_DATA0	U80.H10	U85.2
DDR_G2_DATA1	U80.J10	U85.4
DDR_G2_DATA2	U80.F10	U85.5
DDR_G2_DATA3	U80.G10	U85.7
DDR_G2_DATA4	U80.E10	U85.8
DDR_G2_DATA5	U80.D10	U85.10
DDR_G2_DATA6	U80.C10	U85.11
DDR_G2_DATA7	U80.C11	U85.13
DDR_G2_DATA8	U80.K14	U85.54
DDR_G2_DATA9	U80.L14	U85.56
DDR_G2_DATA10	U80.F14	U85.57
DDR_G2_DATA11	U80.G14	U85.59
DDR_G2_DATA12	U80.D14	U85.60
DDR_G2_DATA13	U80.E14	U85.62
DDR_G2_DATA14	U80.L15	U85.63
DDR_G2_DATA15	U80.K15	U85.65

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_G2_ADD0	U80.G12	U85.29
DDR_FPGA_G2_ADD1	U80.F12	U85.30
DDR_FPGA_G2_ADD2	U80.D12	U85.31
DDR_FPGA_G2_ADD3	U80.L13	U85.32
DDR_FPGA_G2_ADD4	U80.M13	U85.35
DDR_FPGA_G2_ADD5	U80.J13	U85.36
DDR_FPGA_G2_ADD6	U80.K13	U85.37
DDR_FPGA_G2_ADD7	U80.G13	U85.38
DDR_FPGA_G2_ADD8	U80.H13	U85.39
DDR_FPGA_G2_ADD9	U80.E13	U85.40
DDR_FPGA_G2_ADD10	U80.F13	U85.28
DDR_FPGA_G2_ADD11	U80.D13	U85.41
DDR_FPGA_G2_ADD12	U80.C13	U85.42
DDR_FPGA_G2_ADD13	U80.M15	U85.17
DDR_FPGA_G2_UDQS	U80.F15	U85.51
DDR_FPGA_G2_LDQS	U80.F11	U85.16
DDR_FPGA_G2_UDM	U80.H15	U85.47
DDR_FPGA_G2_LDM	U80.H11	U85.20
DDR_FPGA_G2_BA0	U80.F9	U85.26
DDR_FPGA_G2_BA1	U80.E9	U85.27
DDR_FPGA_G2_CASN	U80.H12	U85.22
DDR_FPGA_G2_CKE	U80.K11	U85.44
DDR_FPGA_G2_CSN	U80.K12	U85.24
DDR_FPGA_G2_RASN	U80.J12	U85.23
DDR_FPGA_G2_WEN	U80.L12	U85.21
DDR_I1_DATA0	U79.AN20	U73.2
DDR_I1_DATA1	U79.AM20	U73.4
DDR_I1_DATA2	U79.AP20	U73.5
DDR_I1_DATA3	U79.AR20	U73.7

Signal Name	FPGA Pin	DDR SDRAM
DDR_I1_DATA4	U79.AV19	U73.8
DDR_I1_DATA5	U79.AU19	U73.10
DDR_I1_DATA6	U79.AW19	U73.11
DDR_I1_DATA7	U79.AY19	U73.13
DDR_I1_DATA8	U79.AT18	U73.54
DDR_I1_DATA9	U79.AR18	U73.56
DDR_I1_DATA10	U79.AV17	U73.57
DDR_I1_DATA11	U79.AV18	U73.59
DDR_I1_DATA12	U79.AN18	U73.60
DDR_I1_DATA13	U79.AM18	U73.62
DDR_I1_DATA14	U79.AU17	U73.63
DDR_I1_DATA15	U79.AT17	U73.65
DDR_FPGA_I1_ADD0	U79.AT19	U73.29
DDR_FPGA_I1_ADD1	U79.AR19	U73.30
DDR_FPGA_I1_ADD2	U79.AM19	U73.31
DDR_FPGA_I1_ADD3	U79.AL19	U73.32
DDR_FPGA_I1_ADD4	U79.AP19	U73.35
DDR_FPGA_I1_ADD5	U79.AN19	U73.36
DDR_FPGA_I1_ADD6	U79.AR17	U73.37
DDR_FPGA_I1_ADD7	U79.AP17	U73.38
DDR_FPGA_I1_ADD8	U79.AL18	U73.39
DDR_FPGA_I1_ADD9	U79.AL17	U73.40
DDR_FPGA_I1_ADD10	U79.AM17	U73.28
DDR_FPGA_I1_ADD11	U79.AN17	U73.41
DDR_FPGA_I1_ADD12	U79.AP16	U73.42
DDR_FPGA_I1_ADD13	U79.AN16	U73.17
DDR_FPGA_I1_UDQS	U79.AY18	U73.51
DDR_FPGA_I1_LDQS	U79.AV20	U73.16
DDR_FPGA_I1_UDM	U79.AW17	U73.47

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_I1_LDM	U79.AL21	U73.20
DDR_FPGA_I1_BA0	U79.AW20	U73.26
DDR_FPGA_I1_BA1	U79.AY20	U73.27
DDR_FPGA_I1_CASN	U79.AT16	U73.22
DDR_FPGA_I1_CKE	U79.AW16	U73.44
DDR_FPGA_I1_CSN	U79.AV16	U73.24
DDR_FPGA_I1_RASN	U79.AR16	U73.23
DDR_FPGA_I1_WEN	U79.AL15	U73.21
DDR_I2_DATA0	U79.AW14	U83.2
DDR_I2_DATA1	U79.AV14	U83.4
DDR_I2_DATA2	U79.AM15	U83.5
DDR_I2_DATA3	U79.AN15	U83.7
DDR_I2_DATA4	U79.AU14	U83.8
DDR_I2_DATA5	U79.AT14	U83.10
DDR_I2_DATA6	U79.AN14	U83.11
DDR_I2_DATA7	U79.AM14	U83.13
DDR_I2_DATA8	U79.AM13	U83.54
DDR_I2_DATA9	U79.AL13	U83.56
DDR_I2_DATA10	U79.AP13	U83.57
DDR_I2_DATA11	U79.AN13	U83.59
DDR_I2_DATA12	U79.AR12	U83.60
DDR_I2_DATA13	U79.AP12	U83.62
DDR_I2_DATA14	U79.AT12	U83.63
DDR_I2_DATA15	U79.AU12	U83.65
DDR_FPGA_I2_ADD0	U79.AV15	U83.29
DDR_FPGA_I2_ADD1	U79.AU15	U83.30
DDR_FPGA_I2_ADD2	U79.AY14	U83.31
DDR_FPGA_I2_ADD3	U79.AY15	U83.32
DDR_FPGA_I2_ADD4	U79.AV13	U83.35

Signal Name	FPGA Pin	DDR SDRAM
DDR_FPGA_I2_ADD5	U79.AU13	U83.36
DDR_FPGA_I2_ADD6	U79.AW13	U83.37
DDR_FPGA_I2_ADD7	U79.AY13	U83.38
DDR_FPGA_I2_ADD8	U79.AN12	U83.39
DDR_FPGA_I2_ADD9	U79.AM12	U83.40
DDR_FPGA_I2_ADD10	U79.AV11	U83.28
DDR_FPGA_I2_ADD11	U79.AU11	U83.41
DDR_FPGA_I2_ADD12	U79.AY10	U83.42
DDR_FPGA_I2_ADD13	U79.AY11	U83.17
DDR_FPGA_I2_UDQS	U79.AT13	U83.51
DDR_FPGA_I2_LDQS	U79.AR15	U83.16
DDR_FPGA_I2_UDM	U79.AW12	U83.47
DDR_FPGA_I2_LDM	U79.AR14	U83.20
DDR_FPGA_I2_BA0	U79.AL16	U83.26
DDR_FPGA_I2_BA1	U79.AM16	U83.27
DDR_FPGA_I2_CASN	U79.AV10	U83.22
DDR_FPGA_I2_CKE	U79.AN11	U83.44
DDR_FPGA_I2_CSN	U79.AR11	U83.24
DDR_FPGA_I2_RASN	U79.AW10	U83.23
DDR_FPGA_I2_WEN	U79.AU10	U83.21

7 Rocket IO Transceivers

The multigigabit transceivers (MGTs) can transmit data at speeds from 622 Mb/s up to 3.125 Gb/s (determined be the speed grade of the part, please refer to the Xilinx datasheet). MGTs are capable of various high-speed serial standards such as Gigabit Ethernet, FiberChannel, InfiniBand, and XAUI. In addition, the channel-bonding feature aggregates multiple channels, allowing for even higher data transfer rates. For additional information on RocketIO transceivers, see the *RocketIO Transceiver User Guide* at: http://www.xilinx.com/publications/products/v2pro/userguide/ug024.pdf

The DN6000K10 board has 10 RocketIO transceivers available on the topside of the FPGA and 10 on the bottom side. These 20 transceivers are connected in various configurations depending on the FPGA position on the board; refer to the block

diagram for more information. FPGA A/C/G/I has access to two SMA interfaces, while the rest of the RocketIO interfaces are used for chip-to-chip communication. Refer to the RocketIO Block Diagram in Figure 36.

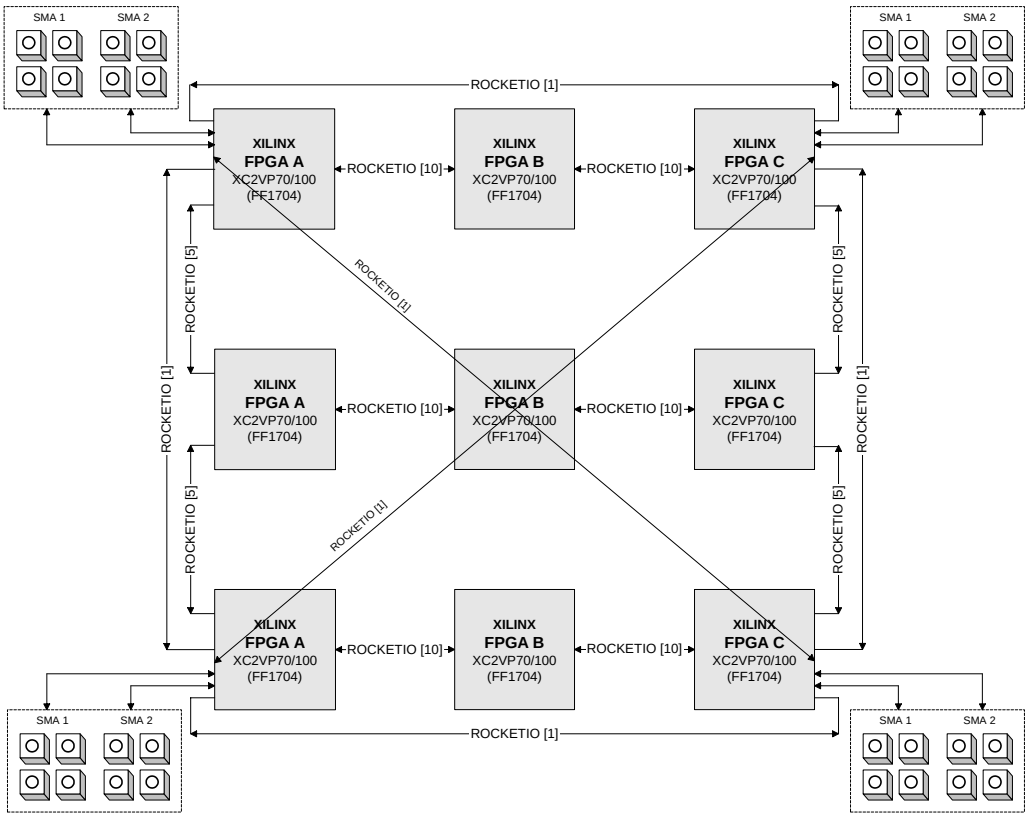


Figure 36 - RocketIO Block Diagram

7.1 SMA Connectors

The SMA connectors allow for direct connection the FPGA MGT interfaces.

7.1.1 FPGA to SMA Connector

The DN6000K10 board provides two discrete MGT channels for FPGA A/C/G/I. The connection between the FPGA and the SMA connectors is fairly simple, involving only one wire per connector, as well as a few capacitors and resistors to AC-couple the signals. These connections are also shown in Table 19.

Table 19 - Connections between FPGA and SMA Connectors

Signal Name	FPGA Pin	Connector
FPGAA_SMA1_TxP	U27.A40	J9
FPGAA_SMA1_TxN	U27.A41	J8

Signal Name	FPGA Pin	Connector
FPGAA_SMA1_RxP	U27.A39	J13
FPGAA_SMA1_RxN	U27.A38	J12
FPGAA_SMA2_TxP	U27.A36	J11
FPGAA_SMA2_TxN	U27.A37	J10
FPGAA_SMA2_RxP	U27.A35	J15
FPGAA_SMA2_RxN	U27.A34	J14
FPGAC_SMA1_TxP	U29.BB4	J27
FPGAC_SMA1_TxN	U29.BB5	J30
FPGAC_SMA1_RxP	U29.BB3	J28
FPGAC_SMA1_RxN	U29.BB2	J31
FPGAC_SMA2_TxP	U29.BB8	J23
FPGAC_SMA2_TxN	U29.BB9	J25
FPGAC_SMA2_RxP	U29.BB7	J24
FPGAC_SMA2_RxN	U29.BB6	J26
FPGAG_SMA1_TxP	U80.A8	J38
FPGAG_SMA1_TxN	U80.A9	J37
FPGAG_SMA1_RxP	U80.A7	J46
FPGAG_SMA1_RxN	U80.A6	J45
FPGAG_SMA2_TxP	U80.A4	J40
FPGAG_SMA2_TxN	U80.A5	J39
FPGAG_SMA2_RxP	U80.A3	J48
FPGAG_SMA2_RxN	U80.A2	J47
FPGAI_SMA1_TxP	U79.BB36	J42
FPGAI_SMA1_TxN	U79.BB37	J41
FPGAI_SMA1_RxP	U79.BB35	J50
FPGAI_SMA1_RxN	U79.BB34	J49
FPGAI_SMA2_TxP	U79.BB40	J44
FPGAI_SMA2_TxN	U79.BB41	J43
FPGAI_SMA2_RxP	U79.BB39	J52

Signal Name	FPGA Pin	Connector
FPGAI_SMA2_RxN	U79.BB38	J51

Please note the RocketIO Transceiver performance in Table 20:

Table 20 - RocketIO Performance

Item	Speed Grade			Units
	-7	-6	-5	
RocketIO Transceiver (FF)	3.125	3.125	2.0	Gb/s
PowerPC Processor Block	400	350	300	MHz

8 CPU Debug and CPU Trace

The DN6000K10 board includes two CPU debugging interfaces for FPGA A/C, the CPU Debug (vertical headers, i.e., JP1 and JP2) and the Combined CPU Trace and Debug, (vertical mictor connector, i.e., J18 and J19). These connectors can be used in conjunction with third party tools, or in some cases the Xilinx Parallel Cable IV, to debug software as it runs on the processor.

The PowerPC™ 405 CPU core includes dedicated debug resources that support a variety of debug modes for debugging during hardware and software development. These debug resources include:

- Internal debug mode for use by ROM monitors and software debuggers
- External debug mode for use by JTAG debuggers
- Debug wait mode, which allows the servicing of interrupts while the processor appears to be stopped
- Real-time trace mode, which supports event triggering for real-time tracing

Debug modes and events are controlled using debug registers in the processor. The debug registers are accessed either through software running on the processor or through the JTAG port. The debug modes, events, controls, and interfaces provide a powerful combination of debug resources for hardware and software development tools.

The JTAG port interface supports the attachment of external debug tools, such as the ChipScope™ Integrated Logic Analyzer, a powerful tool providing logic analyzer capabilities for signals inside an FPGA, without the need for expensive external

instrumentation. Using the JTAG test access port, a debug tool can single-step the processor and examine the internal processor state to facilitate software debugging. This capability complies with the IEEE 1149.1 specification for vendor-specific extensions and is, therefore, compatible with standard JTAG hardware for boundary-scan system testing.

8.1 CPU Debug

External-debug mode can be used to alter normal program execution. It provides the ability to debug system hardware as well as software. The mode supports multiple functions: starting and stopping the processor, single-stepping instruction execution, setting breakpoints, as well as monitoring processor status. Access to processor resources is provided through the CPU Debug port.

The PPC405 JTAG (Joint Test Action Group) Debug port complies with IEEE standard 1149.1-1990, IEEE Standard Test Access Port and Boundary Scan Architecture. This standard describes a method for accessing internal chip resources using a four-signal or five-signal interface. The PPC405 JTAG Debug port supports scan-based board testing and is further enhanced to support the attachment of debug tools. These enhancements comply with the IEEE 1149.1 specifications for vendor-specific extensions and are compatible with standard JTAG hardware for boundary-scan system testing.

The PPC405 JTAG debug port supports the four required JTAG signals: TCK, TMS, TDI, and TDO. It also implements the optional TRST signal. The frequency of the JTAG clock signal can range from 0 MHz (DC) to one-half of the processor clock frequency. The JTAG debug port logic is reset at the same time the system is reset, using TRST. When TRST is asserted, the JTAG TAP controller returns to the test-logic reset state.

Refer to the *PPC405 Processor Block Manual* for more information on the JTAG debug-port signals. Information on JTAG is found in the IEEE standard 1149.1-1990.

8.1.1 CPU Debug Connectors

Figure 37 shows JP1, the vertical header used to debug the operation of software in the PPC of FPGA A (there is another connector on FPGA C). This is done using debug tools such as Parallel Cable IV or third party tools. This connector cannot be used when the Mictor connector is in use.

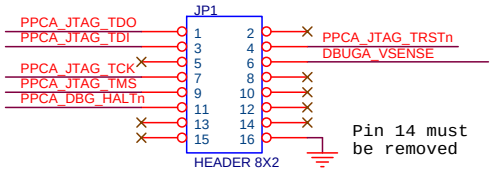


Figure 37 - CPU Debug Connector

8.1.2 CPU Debug Connection to FPGA's

The connection between the PPC debug connectors and the FPGA's are shown in Table 21. These signals are attached to the PowerPC™ 405 JTAG debug resources using normal FPGA routing resources. The JTAG debug resources are not hard-wired to particular pins, and are available for attachment in the FPGA fabric, making it is possible to route these signals to whichever FPGA pins the user would prefer to use.

Table 21 - CPU Debug connection to FPGA

Signal Name	FPGA Pin	Connector
PPCA_JTAG_TDO	U27.E36	JP1.1
PPCA_JTAG_TDI	U27.D36	JP1.3
PPCA_JTAG_TRSTn	GND	JP1.4
PPCA_JTAG_TCK	U27.D37	JP1.7
PPCA_JTAG_TMS	U27.E37	JP1.7
PPCA_DBG_HALTn	U27.F36	JP1.7
PPCC_JTAG_TDO	U29.AC34	JP2.1
PPCC_JTAG_TDI	U29.AC33	JP2.3
PPCC_JTAG_TRSTn	GND	JP2.4
PPCC_JTAG_TCK	U29.AD33	JP2.7
PPCC_JTAG_TMS	U29.AD34	JP2.7
PPCC_DBG_HALTn	U29.AC36	JP2.7

8.2 CPU Trace

The CPU Trace port accesses the real-time, trace-debug capabilities built into the PowerPC™ 405 CPU core. Real-time trace-debug mode supports real-time tracing of the instruction stream executed by the processor. In this mode, debug events are used to cause external trigger events. An external trace tool uses the trigger events to control the collection of trace information. The broadcast of trace information occurs independently of external trigger events (trace information is always supplied by the processor).

Real-time trace-debug does not affect processor performance. Real-time trace-debug mode is always enabled. However, the trigger events occur only when both internal-debug mode and external debug mode are disabled. Most trigger events are blocked when either of those two debug modes is enabled. Information on the trace-debug capabilities, how trace-debug works, and how to connect an external trace tool is available in the *RISCWatch Debugger User's Guide*.

8.2.1 CPU Trace Connectors

Figure 38 shows J18, the vertical header used to trace the operation of software in the PPC of FPGA A (there is another connector on FPGA C). Agilent/Windriver has defined a Trace Port Analyzer (TPA) port for the PowerPC 4xx line of CPU cores that combines the CPU Trace and the CPU Debug interfaces onto a single 38-pin Mictor connector. This provides for high-speed, controlled-impedance signaling.

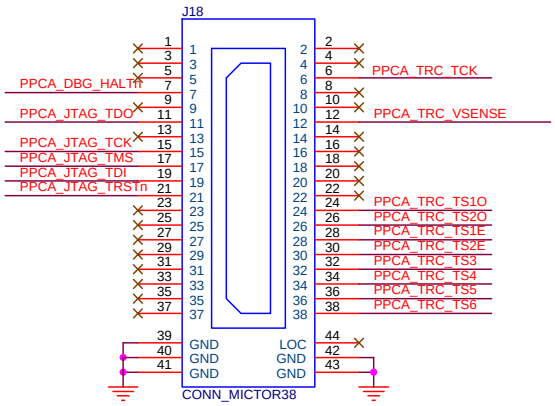


Figure 38 - Combined Trace/Debug Connector Pinout

8.2.2 Combined CPU Trace/Debug Connection to FPGA's

The connection between the Combined CPU Trace and Debug Port connectors is shown in Table 22. The connections to the FPGA are shared with the CPU Trace and CPU Debug interfaces discussed in previous sections.

Table 22 - Combined CPU Trace/Debug connection to FPGA

Signal Name	FPGA Pin	Connector
PPCA_TRC_TCK	U27.T37	J18.6
PPCA_DBG_HALTn	U27.F36	J18.7
PPCA_TRC_VSENSE	N.A.	J18.12
PPCA_JTAG_TDO	U27.E36	J1711
PPCA_JTAG_TCK	U27.D37	J18.15
PPCA_JTAG_TMS	U27.E37	J18.17
PPCA_JTAG_TDI	U27.D36	J18.19
PPCA_JTAG_TRSTn	GND	J18.21
PPCA_TRC_TS1O	U27.R37	J18.24
PPCA_TRC_TS2O	U27.P37	J18.26
PPCA_TRC_TS1E	U27.N37	J18.28

Signal Name	FPGA Pin	Connector
PPCA_TRC_TS2E	U27.L37	J18.30
PPCA_TRC_TS3	U27.K37	J18.32
PPCA_TRC_TS4	U27.J37	J18.34
PPCA_TRC_TS5	U27.H37	J18.36
PPCA_TRC_TS6	U27.G37	J18.38
PPCC_TRC_TCK	U29.AN37	J19.6
PPCC_DBG_HALTn	U29.AC36	J19.7
PPCC_TRC_VSENSE	N.A.	J19.12
PPCC_JTAG_TDO	U29.AC34	J1811
PPCC_JTAG_TCK	U29.AD33	J19.15
PPCC_JTAG_TMS	U29.AD34	J19.17
PPCC_JTAG_TDI	U29.AC33	J19.19
PPCC_JTAG_TRSTn	GND	J19.21
PPCC_TRC_TS1O	U29.AM37	J19.24
PPCC_TRC_TS2O	U29.AK37	J19.26
PPCC_TRC_TS1E	U29.AJ37	J19.28
PPCC_TRC_TS2E	U29.AH37	J19.30
PPCC_TRC_TS3	U29.AG37	J19.32
PPCC_TRC_TS4	U29.AF37	J19.34
PPCC_TRC_TS5	U29.AD37	J19.36
PPCC_TRC_TS6	U29.AC37	J19.38

9 GPIO LED's

9.1 Status Indicators

The DN6000K10 uses DS1 and DS2 to visually indicate the status of the board. DS1 is controller by the MCU (U7) and the Configuration FPGA (U13) controls DS2.

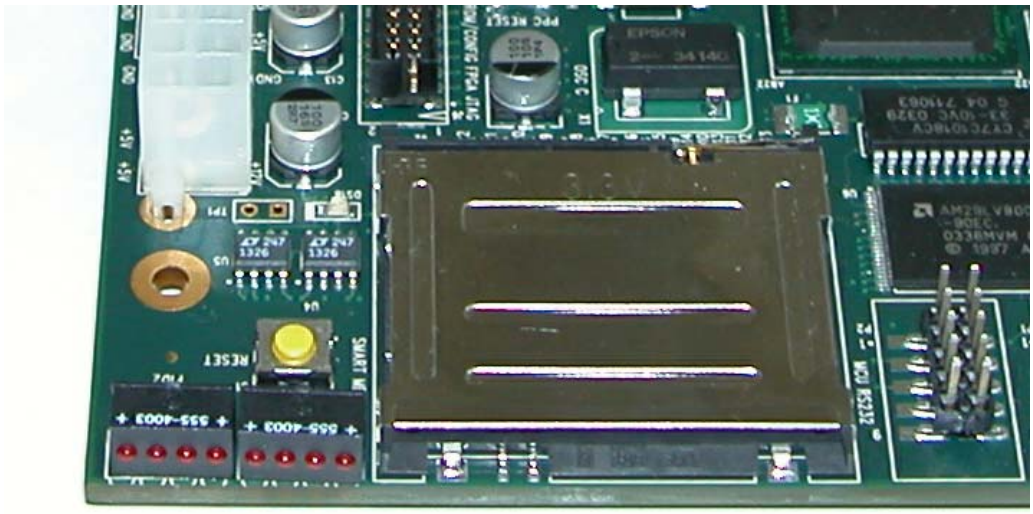


Table 23 lists the function of the CPLD LED's. The LED's is number from left to right CPLD_LED0n to CPLD_LED3n.

Table 23 - CPLD LED's

Signal Name	Device	LED	Description
CFPGA_LEDn0	U13.L1	DS2.1	Blinks when configuring over USB
CFPGA_LEDn1	U13.L5	DS2.2	Blinks when reading data from the SmartMedia card
CFPGA_LEDn2	U13.L4	DS2.3	Blinks when MCU is reading/writing data to/from the FPGA's
CFPGA_LEDn3	U13.L3	DS2.4	Blinks when the MCU is read/writing data to/from the FPGA's via the USB interface

The MCU_LED's are used to show which FPGA is currently being configured (either by SmartMedia or over USB), and also give the user overall configuration status.

Table 24 - MCU LED's

FPGA / Status	MCU_LED0n	MCU_LED1n	MCU_LED2n	MCU_LED3n
A	On	Off	Off	off
B	Off	On	Off	Off
C	On	On	Off	Off
D	Off	Off	On	Off

FPGA / Status	MCU_LED0n	MCU_LED1n	MCU_LED2n	MCU_LED3n
E	On	Off	On	Off
F	Off	On	On	Off
G	On	On	On	Off
H	Off	Off	Off	On
I	On	Off	Off	On
Successful Configuration	Off	Off	On	On
Error during Configuration / No FPGAs configured	Blink	Blink	Blink	Blink

9.2 FPGA A GPIO LED's

The DN6000K10 provides 10 GPIO LED's directly connected to FPGA A IO Bank 2 pins. Table 25 lists the FPGA GPIO LED's on the DN6000K10 and is available to the user. The signals are active LOW.

Table 25 – FPGA A GPIO LED's

Signal Name	FPGA A	LED
LED0	U29.P11	DS8
LED1	U27.P12	DS9
LED2	U27.R11	DS10
LED3	U27.R12	DS11
LED4	U27.T11	DS12
LED5	U27.T12	DS13
LED6	U27.U12	DS14
LED7	U27.V11	DS15
LED8	U27.U11	DS16
LED9	U27.V12	DS17

10 Power System

The DN6000K10 supports a wide range of technologies, from legacy devices like serial ports, to DDR SDRAM and RocketIO multi-gigabit transceivers (MGTs). This wide range of technologies requires a wide range of power supplies. These are provided on the DN6000K10 using a combination of switching and linear power regulators.

10.1 Stand Alone Operation

An external ATX power supply is used to supply power to the DN6000K10 (refer to Figure 39). The external power supply connects to header P16, Molex type header P/N 39-29-9202.

The DN6000K10 has the following power supplies:

- +1.25V
- +1.5V
- +2.5V
- +3.3V
- +5V
- +12V

The +1.5V, +2.5V power supplies are generated from the +5V supply on the External ATX power supply, while +3.3V comes directly from the ATX power supply.

Figure 39 - ATX Power Supply

Any ATX type power supply is adequate. The Dini Group recommends a power supply rated for 250W. Note: The switching regulators in the Power Supply may require and external load to operate within specifications (the DN6000K10 may not meet the minimum load requirements). The Dini Group recommends attaching an old disk drive to one of the spare connectors.

10.1.1 External Power Connector

Figure 40 indicates the connections to the external power connector. This header is fully polarized to prevent reverse connection and is rated for 1500VAC at 6A per contact.

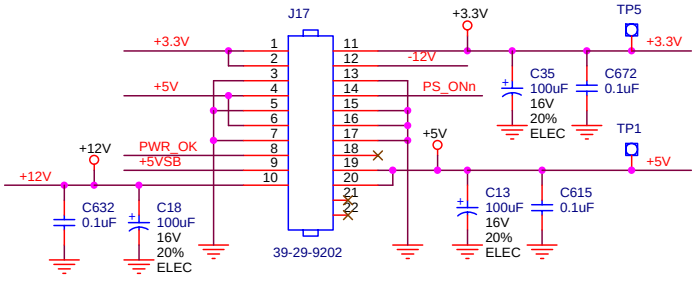


Figure 40 - External Power Connection

Note: Header J17 is not hot-plug able. Do not attach power while power supply is ON.

Since the DN6000K10 is populated with up to 9 Virtex-II Pro FPGA's, depending on the RTL design, significant power demands may be placed on the on-board +1.5V/+2.5V switching power supplies. Optional PWR connectorS allows for connection to a high power external PSU's. These connectors would need to be used in place of the +1.5V/+2.5V switching supplies, PSU1/PSU2. A Molex connector, P/N 42819-4212, is used and is rated at 600V/48A (refer to Figure 41).

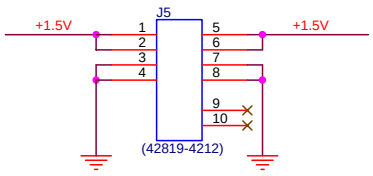


Figure 41 - Optional PWR Connector

10.1.2 Power Monitors

Power supply monitor (U5) is used to monitor the +1.5V, +2.5V, +3.3V, and +5V supplies (for more information on these devices, please refer to the datasheet for the LT2900 from Linear Technology). The power supply monitor also provides a push-button reset input that is utilized to reset the various sub-circuits of the DN6000K10. After power-up, SYS_RSTn remains asserted for approximately 10ms.

10.1.3 Power Indicators

There are six LED's on the DN6000K10 used to indicates the presence of the following voltage sources (refer to Table 26):

Table 26 – Voltage Indicators

Voltage Source	LED
PWR_OK	DS18
+2.5V	DS3
+3.3V	DS4
+5V	DS5
+12V	DS6
-12V	DS7

10.1.4 Front Panel Indicator/Switch

In order to power the board from the ATX Power Supply, PS_ONn needs to be pulled to GND. There is also a PWR_ON_LED signal to indicate the power has been turned on (refer to Figure 42).

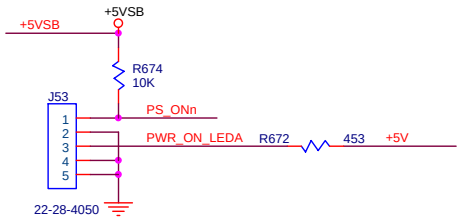


Figure 42 - Front Panel Indicator/Switch

11 Test Header & Daughter Card Connections

11.1 Test Header

The DN6000K10 offers three 200-pin test headers (P9, P10, P11) that allow the user connection to discrete FPGA pins, refer to Figure 43, Test Header A is shown:

Note: Use of a Duaghter card requires the FPGA fan to be removed, leaving the heatsink in place.

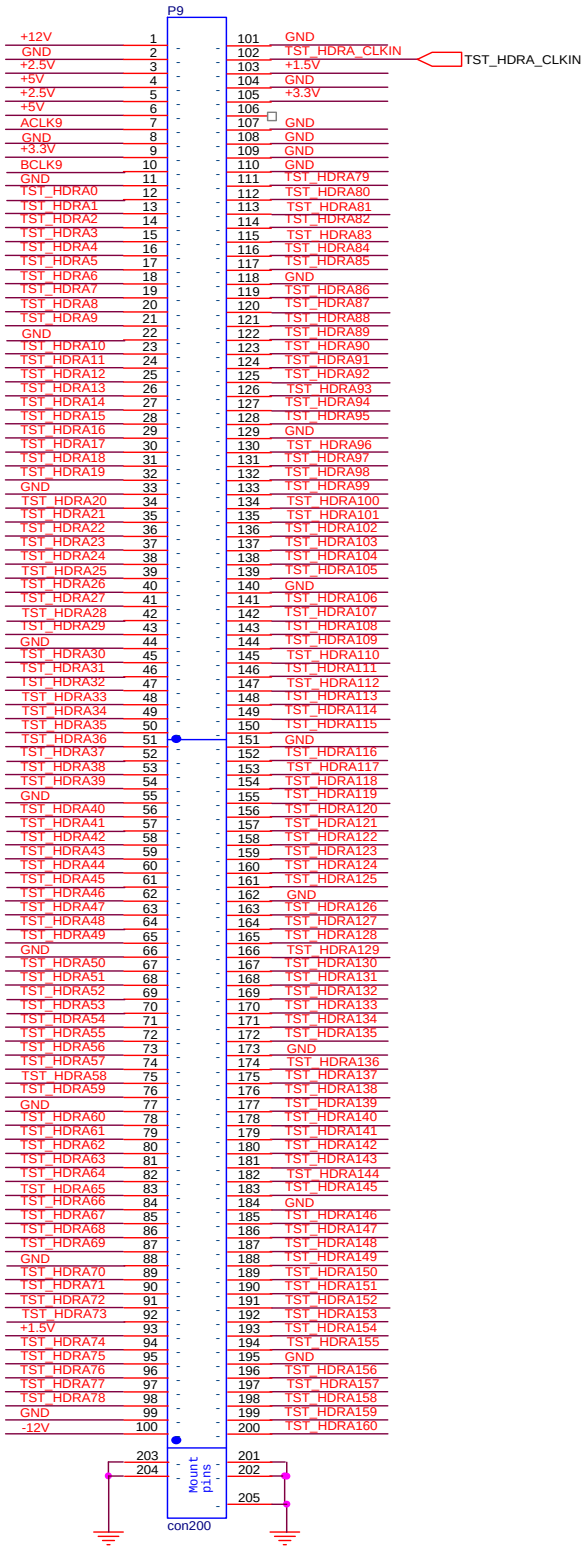


Figure 43 - Test Header

11.1.1 Test Header Connector

Micropax connector (200 pin) is used as a standard interface to all the Dini Group logic emulation boards. This connector has a specified current rating of 0.5 amps per contact. See datasheet for more information P/N 91294-003.

11.1.2 Test Header Pin Numbering

Figure 44 indicates the pin numbering scheme used on the test headers.

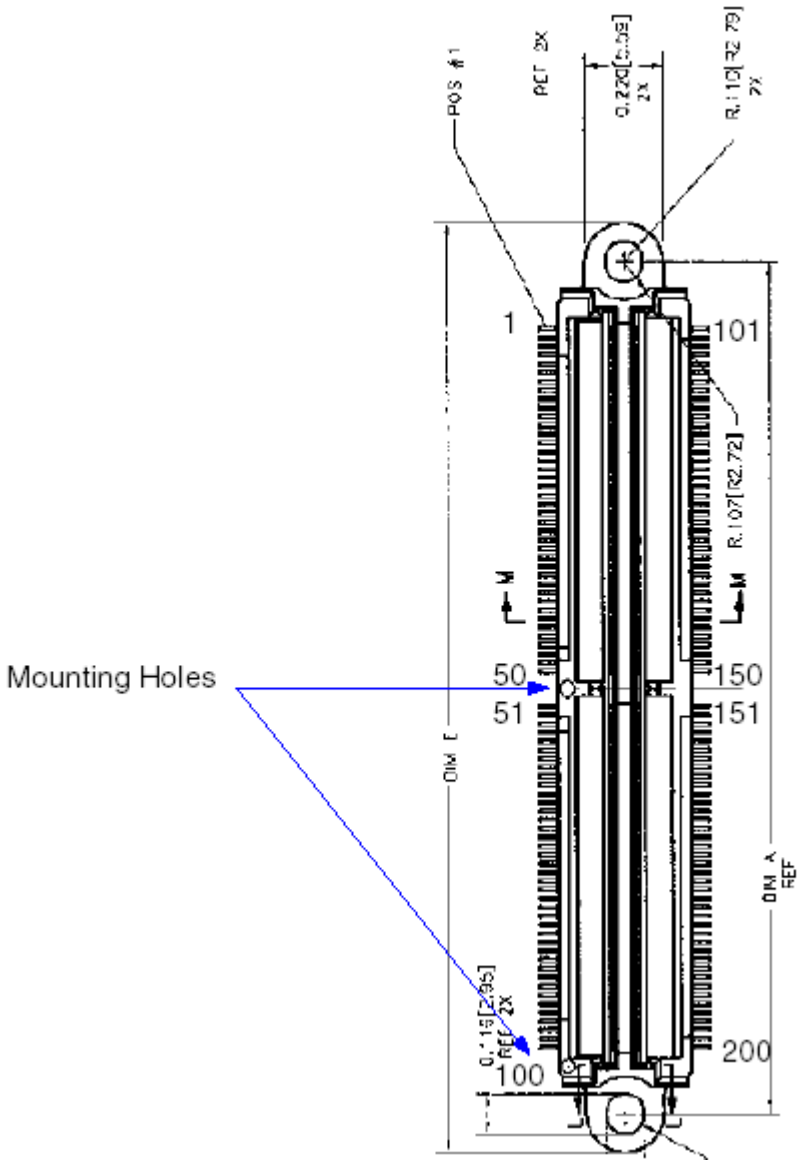


Figure 44 - Test Header Pin Numbering

11.2 DN3000K10SD Daughter Card

The Dini Group manufactures a daughter "DN3000K10SD" card that allows the user connection to the FPGA IO pins. The daughter card has the following features:

- Buffered I/O, Passive and Active Bus Drivers
- Unbuffered I/O
- Differential LVDS pairs (Note: Not available on DN6000K10 Logic Emulation board)
- Headers for Test Points

The daughter card contains headers that may be useful with certain types of oscilloscope probes, or when wiring pins to prototype areas. Figure 45 is a block diagram of the daughter card.

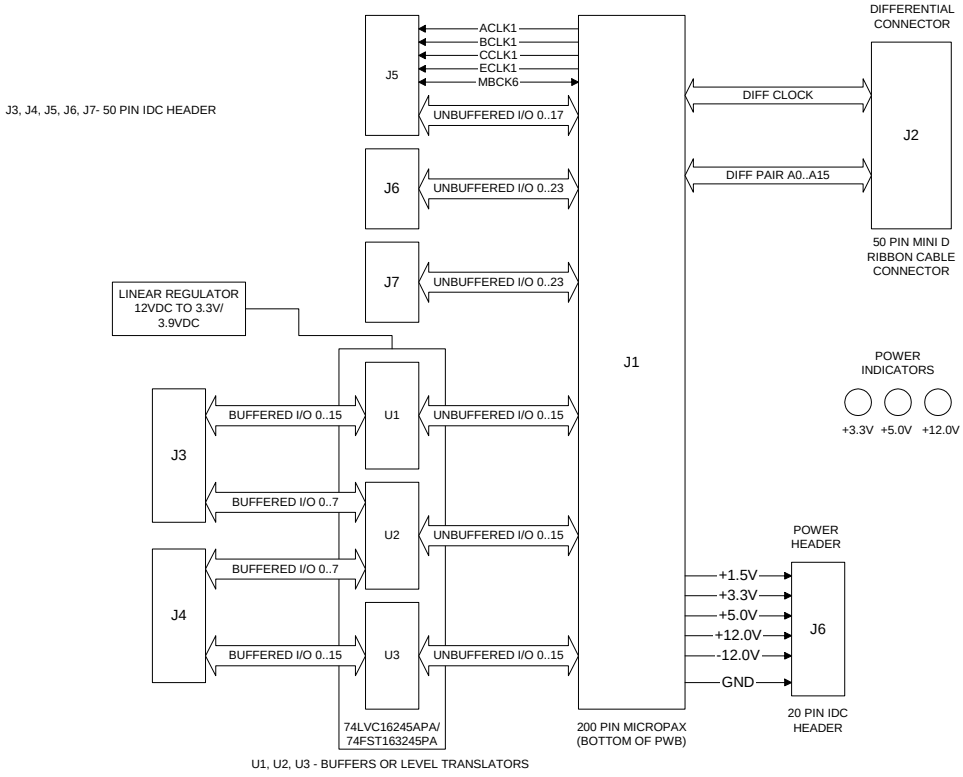


Figure 45 - DN3000K10SD Daughter Card Block Diagram

The DN3000K10SD Daughter Card provides 16-differential pairs, 48-buffered (passive/active) I/O, and 66-unbuffered I/O signals. The DN3000K10SD Daughter Card is pictured in Figure 46.

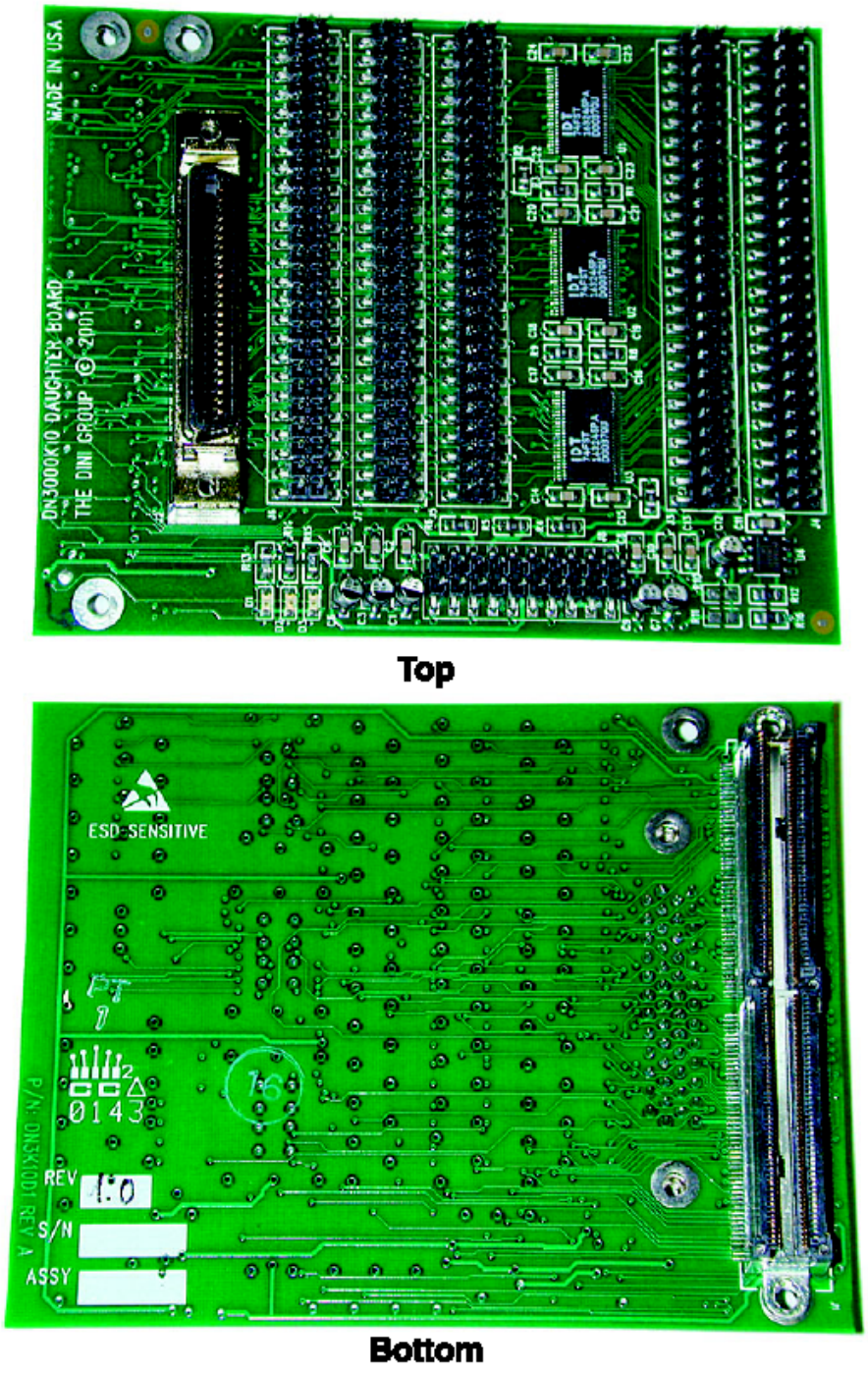


Figure 46 - DN3000K10S Daughter Card

Figure 47 show the assembly drawing of the DN3000K10SD Daughter Card. IDT74FST163245 devices (U1, U2, U3) are used as bus switches in the passive mode,

and the IDT74LVC16245A (U1, U2, U3) devices are used as bus transceivers in the active mode. The DN3000K10SD has separate enable/direction signals for each driver.

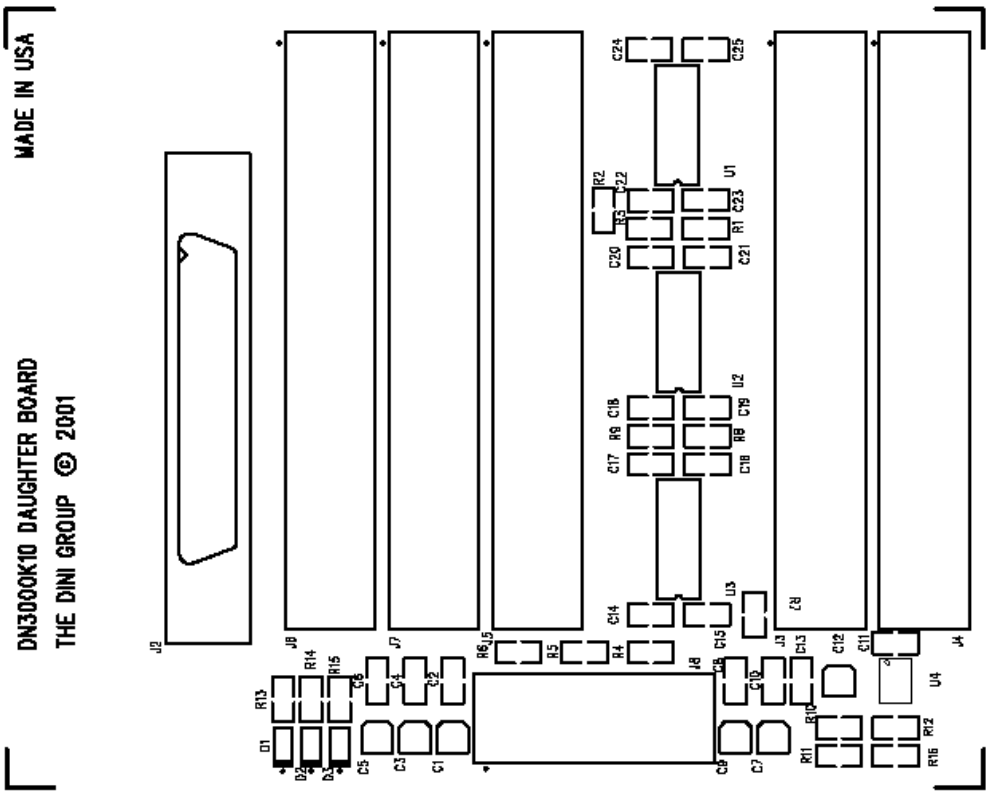


Figure 47 - Assembly drawing for the DN3000K10SD

NOTE: Signals P4NX7 and P4NX6 are also used for direction select and output enable on U2 and U3 respectively.

11.2.1 Daughter Card LED's

The LED's act as visual indicators, representing the presence of active power sources.

- D1 - LED indicating +3.3 V present
- D2 - LED indicating +5.0 V present
- D3 - LED indicating +12 V present

Under normal operating conditions, all LED's should be ON.

11.2.2 Power Supply

A linear power supply (U4) is present to provide level shift/translation functions when the board is populated with passive bus switches. Resistors R10 and R11 can be used to select alternate voltage sources, +5V or +3.3V, respectively. When used, U4 must be removed in order to prevent contention. The power supplies is rated as follows:

- +5 V power supply is rated for 1 A
- +3.3 V power supply is rated for 1 A
- +1.5 V power supply is rated for 1 A
- +12 V power supply is rated for 0.5 A
- –12 V power supply is rated for 0.5 A

NOTE: Never populate R10/R11 simultaneously, this will result in a shorting the +3.3V and +5V power supplies.

Header J8 allows external connection to the Power Sources (refer to Table 27 for connection details).

Table 27 - External Power Connections

Pin	Function	Pin	Function
J8.1	GND	J8.11	GND
J8.2	+5V	J8.12	+1.5V
J8.3	GND	J8.13	GND
J8.4	+5V	J8.14	+12V
J8.5	GND	J8.15	GND
J8.6	+3.3V	J8.16	+12V
J8.7	GND	J8.17	GND
J8.8	+3.3V	J8.18	-12V
J8.9	GND	J8.19	GND
J8.10	+1.5V	J8.20	-12V

11.2.3 Unbuffered IO

The DN3000k10SD Daughter Card provides 66-unbuffered I/O signals, including 5 single ended clock signals available on headers J5, J6, and J7. The function of these signals is position dependent.

11.2.4 Buffered IO

The DN3000k10SD Daughter Card provides 48-buffered I/O signals available on headers J3, and J4. The function of these signals is position dependent. U1, U2, and U3 allow for different populating options, and devices can be active or passive:

Active - The LCV162245A is used for asynchronous communication between data buses. It allows data transmission from the A to the B or from the B to the A bus, depending on the logic level at the direction-control (DIR) input. The output-enable (OE#) input can be used to disable the device so that the busses are effectively isolated

Passive - The FST163245 bus switches are used to connect or isolate two ports without providing any current sink or source capabilities. Thus, they generate little or no noise of their own while providing a low resistance path for an external driver. The output-enable (OE#) input can be used to disable the device so that the busses are effectively isolated.

11.2.5 LVDS IO

Low-voltage differential signaling (LVDS) is a signaling method used for high-speed transmission of binary data over copper. It is well recognized that the benefits of balanced data transmission begin to outweigh the costs over single-ended techniques when the signal transmission times approach 10 ns. This represents signaling rates of about 30 Mbps or clock rates of 60 MHz (in single-edge clocking systems) and above. LVDS is defined in the TIA/EIA-644 standards.

Connector J2 is a Mini D Ribbon (MDR) connector (50-pin) manufactured by 3M, used specifically for high speed LVDS signaling. The connector mates with a standard off-the-shelf 3M-cable assembly:

P/N 14150-EZBB-XXX-0LC

where XXX is: 050 = 0.5 m

150 = 1.5 m

300 = 3.0 m

500 = 5.0 m

Please contact 3M for further details: http://www.3m.com/

11.2.6 Connection between FPGA and the Daughter Card Headers

Table 28 shows the IO connections between the DN3000K10SD headers and the FPGA IO pins. The VCCO of the IO banks are connected to +2.5V.

Table 28 - Connection between FPGA and the Daughter Card Headers

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	**Signal Name**	**Connector**	**Test Header**	**Signal Name**	**FPGA Pin**
J1.001	No Connect		P9.1	+12V	
J1.002	No Connect		P9.2	GND	
J1.003	ACLK1	J5.1	P9.3	+2.5V	
J1.004	No Connect		P9.4	+5V	
J1.005	BCLK1	J5.3	P9.5	+2.5V	
J1.006	No Connect		P9.6	+5V	
J1.007	CCLK1	J5.5	P9.7	ACLK9	
J1.008	No Connect		P9.8	GND	
J1.009	No Connect		P9.9	+3.3V	
J1.010	BP2N3(P2N3)	J3.1	P9.10	BCLK9	
J1.011	No Connect		P9.11	GND	
J1.012	BP2N2(P2N2)	J3.3	P9.12	TST_HDRA0	U28.AF41
J1.013	P2N1	J2.8	P9.13	TST_HDRA1	U28.AE41
J1.014	P2N0	J2.9	P9.14	TST_HDRA2	U28.AD41
J1.015	BP2NX7(P2NX7)	J3.5	P9.15	TST_HDRA3	U28.AC31
J1.016	BP2NX6(P2NX6)	J3.7	P9.16	TST_HDRA4	U28.AC33
J1.017	BP2NX5(P2NX5)	J3.9	P9.17	TST_HDRA5	U28.AC36
J1.018	BP2NX4(P2NX4)	J3.11	P9.18	TST_HDRA6	U28.AC37
J1.019	P2NX1	J2.10	P9.19	TST_HDRA7	U28.AC40
J1.020	P2NX0	J2.11	P9.20	TST_HDRA8	U28.AB33
J1.021	P3NX9	J2.40	P9.21	TST_HDRA9	U28.AB36

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.022	No Connect		P9.22	GND	
J1.023	P3NX8	J2.41	P9.23	TST_HDRA10	U28.AB40
J1.024	BP3NX5(P3NX5)	J3.13	P9.24	TST_HDRA11	U27.AW42
J1.025	BP3NX4(P3NX4)	J3.15	P9.25	TST_HDRA12	U27.AV42
J1.026	BP3N89(P3N89)	J3.17	P9.26	TST_HDRA13	U27.AU42
J1.027	BP3N88(P3N88)	J3.19	P9.27	TST_HDRA14	U27.AT42
J1.028	BP3N87(P3N87)	J3.21	P9.28	TST_HDRA15	U27.AP41
J1.029	BP3N86(P3N86)	J3.23	P9.29	TST_HDRA16	U27.AP42
J1.030	BP3N83(P3N83)	J3.25	P9.30	TST_HDRA17	U27.AN42
J1.031	BP3N82(P3N82)	J3.27	P9.31	TST_HDRA18	U27.AM42
J1.032	BP3N77(P3N77)	J3.29	P9.32	TST_HDRA19	U27.AK41
J1.033	No Connect		P9.33	GND	
J1.034	BP3N76(P3N76)	J3.31	P9.34	TST_HDRA20	U27.AJ42
J1.035	BP3N75(P3N75)	J3.33	P9.35	TST_HDRA21	U27.AH42
J1.036	BP3N74(P3N74)	J3.35	P9.36	TST_HDRA22	U27.AF41
J1.037	P3N69	J2.42	P9.37	TST_HDRA23	U27.AE41
J1.038	P3N68	J2.43	P9.38	TST_HDRA24	U27.AD41
J1.039	BP3N67(P3N67)	J3.37	P9.39	TST_HDRA25	U27.AC33
J1.040	BP3N66(P3N66)	J3.39	P9.40	TST_HDRA26	U27.AC34
J1.041	BP3N63(P3N63)	J3.41	P9.41	TST_HDRA27	U27.AC37
J1.042	BP3N62(P3N62)	J3.43	P9.42	TST_HDRA28	U27.AC40
J1.043	BP3N57(P3N57)	J3.45	P9.43	TST_HDRA29	U27.AB34
J1.044	No Connect		P9.44	GND	
J1.045	BP3N56(P3N56)	J3.47	P9.45	TST_HDRA30	U27.AB39
J1.046	No Connect		P9.46	TST_HDRA31	U27.AB31
J1.047	No Connect		P9.47	TST_HDRA32	U27.AA33

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.048	BP3N49(P3N49)	J4.1	P9.48	TST_HDRA33	U27.AA36
J1.049	BP3N48(P3N48)	J4.3	P9.49	TST_HDRA34	U27.Y32
J1.050	P3N47	J2.19	P9.50	TST_HDRA35	U27.Y34
J1.051	P3N46	J2.20	P9.51	TST_HDRA36	U27.Y36
J1.052	BP3N43(P3N43)	J4.5	P9.52	TST_HDRA37	U27.W31
J1.053	BP3N42(P3N42)	J4.7	P9.53	TST_HDRA38	U27.W33
J1.054	BP3N39(P3N39)	J4.9	P9.54	TST_HDRA39	U27.W35
J1.055	No Connect		P9.55	GND	
J1.056	BP3N38(P3N38)	J4.11	P9.56	TST_HDRA40	U27.W41
J1.057	BP3N35(P3N35)	J4.13	P9.57	TST_HDRA41	U27.V31
J1.058	BP3N34(P3N34)	J4.15	P9.58	TST_HDRA42	U27.V33
J1.059	BP3N29(P3N29)	J4.17	P9.59	TST_HDRA43	U27.V36
J1.060	BP3N28(P3N28)	J4.19	P9.60	TST_HDRA44	U27.V42
J1.061	BP3N27(P3N27)	J4.21	P9.61	TST_HDRA45	U27.U32
J1.062	BP3N26(P3N26)	J4.23	P9.62	TST_HDRA46	U27.U33
J1.063	P3N23	J2.21	P9.63	TST_HDRA47	U27.U35
J1.064	P3N22	J2.22	P9.64	TST_HDRA48	U27.U37
J1.065	BP3N19(P3N19)	J4.25	P9.65	TST_HDRA49	U27.U42
J1.066	No Connect		P9.66	GND	
J1.067	BP3N18(P3N18)	J4.27	P9.67	TST_HDRA50	U27.T33
J1.068	BP3N15(P3N15)	J4.29	P9.68	TST_HDRA51	U27.T36
J1.069	BP3N14(P3N14)	J4.31	P9.69	TST_HDRA52	U27.R31
J1.070	P3N9	J2.23	P9.70	TST_HDRA53	U27.R33
J1.071	P3N8	J2.24	P9.71	TST_HDRA54	U27.R35
J1.072	BP3N7(P3N7)	J4.33	P9.72	TST_HDRA55	U27.R42
J1.073	BP3N6(P3N6)	J4.35	P9.73	TST_HDRA56	U27.P31

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.074	BP3N3(P3N3)	J4.37	P9.74	TST_HDRA57	U27.P33
J1.075	BP3N2(P3N2)	J4.39	P9.75	TST_HDRA58	U27.P35
J1.076	BP4N27(P4N27)	J4.41	P9.76	TST_HDRA59	U27.P41
J1.077	No Connect		P9.77	GND	
J1.078	BP4N26(P4N26)	J4.43	P9.78	TST_HDRA60	U27.N32
J1.079	BP4N21(P4N21)	J4.45	P9.79	TST_HDRA61	U27.N34
J1.080	BP4N20(P4N20)	J4.47	P9.80	TST_HDRA62	U27.N36
J1.081	No Connect		P9.81	TST_HDRA63	U27.N42
J1.082	No Connect		P9.82	TST_HDRA64	U27.M32
J1.083	No Connect		P9.83	TST_HDRA65	U27.M34
J1.084	No Connect		P9.84	TST_HDRA66	U27.M35
J1.085	No Connect		P9.85	TST_HDRA67	U27.M41
J1.086	No Connect		P9.86	TST_HDRA68	U27.L34
J1.087	No Connect		P9.87	TST_HDRA69	U27.L36
J1.088	No Connect		P9.88	GND	
J1.089	No Connect		P9.89	TST_HDRA70	U27.K34
J1.090	No Connect		P9.90	TST_HDRA71	U27.K36
J1.091	No Connect		P9.91	TST_HDRA72	U27.K42
J1.092	No Connect		P9.92	TST_HDRA73	U27.J36
J1.093	No Connect		P9.93	+1.5V	
J1.094	No Connect		P9.94	TST_HDRA74	U27.H41
J1.095	P4NX7	J7.45	P9.95	TST_HDRA75	U27.G42
J1.096	P4NX6	J7.47	P9.96	TST_HDRA76	U27.F42
J1.097	No Connect		P9.97	TST_HDRA77	U27.E42
J1.098	No Connect		P9.98	TST_HDRA78	U27.D42
J1.099	No Connect		P9.99	GND	

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.100	No Connect		P9.100	-12V	
J1.101	No Connect		P9.101	GND	
J1.102	MBCK1	J2.27	P9.102	TST_HDRA_CLKIN	U27.AT22
J1.103	No Connect		P9.103	+1.5V	
J1.104	MBCK0	J2.28	P9.104	GND	
J1.105	No Connect		P9.105	+3.3V	
J1.107	No Connect		P9.107	GND	
J1.108	ECLK1	J5.7	P9.108	GND	
J1.109	No Connect		P9.109	GND	
J1.110	No Connect		P9.110	GND	
J1.111	P2N5	J5.15	P9.111	TST_HDRA79	U28.AH42
J1.112	P2N4	J5.17	P9.112	TST_HDRA80	U28.AG41
J1.113	P2NX11	J2.2	P9.113	TST_HDRA81	U28.AF42
J1.114	P2NX10	J2.1	P9.114	TST_HDRA82	U28.AE42
J1.115	P2NX9	J5.19	P9.115	TST_HDRA83	U28.AD42
J1.116	P2NX8	J5.21	P9.116	TST_HDRA84	U28.AC32
J1.117	P2NX3	J5.23	P9.117	TST_HDRA85	U28.AC34
J1.118	No Connect		P9.118	GND	
J1.119	P2NX2	J5.25	P9.119	TST_HDRA86	U28.AC39
J1.120	P3NX11	J2.29	P9.120	TST_HDRA87	U28.AB31
J1.121	P3NX10	J2.30	P9.121	TST_HDRA88	U28.AB34
J1.122	P3NX7	J2.31	P9.122	TST_HDRA89	U28.AB37
J1.123	P3NX6	J2.32	P9.123	TST_HDRA90	U28.AB39
J1.124	P3NX3	J5.27	P9.124	TST_HDRA91	U27.AW41
J1.125	P3NX2	J5.29	P9.125	TST_HDRA92	U27.AV41
J1.126	P3NX1	J5.31	P9.126	TST_HDRA93	U27.AU41

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.127	P3NX0	J5.33	P9.127	TST_HDRA94	U27.AT41
J1.128	P3N85	J5.35	P9.128	TST_HDRA95	U27.AR41
J1.129	No Connect		P9.129	GND	
J1.130	P3N84	J5.37	P9.130	TST_HDRA96	U27.AN41
J1.131	P3N81	J5.39	P9.131	TST_HDRA97	U27.AM41
J1.132	P3N80	J5.41	P9.132	TST_HDRA98	U27.AL41
J1.133	P3N79	J2.3	P9.133	TST_HDRA99	U27.AK42
J1.134	P3N78	J2.4	P9.134	TST_HDRA100	U27.AJ41
J1.135	P3N73	J2.6	P9.135	TST_HDRA101	U27.AH41
J1.136	P3N72	J2.7	P9.136	TST_HDRA102	U27.AG41
J1.137	P3N71	J2.33	P9.137	TST_HDRA103	U27.AF42
J1.138	P3N70	J2.34	P9.138	TST_HDRA104	U27.AE42
J1.139	P3N65	J5.43	P9.139	TST_HDRA105	U27.AD42
J1.140	No Connect		P9.140	GND	
J1.141	P3N64	J5.45	P9.141	TST_HDRA106	U27.AC36
J1.142	P3N61	J5.47	P9.142	TST_HDRA107	U27.AC39
J1.143	P3N60	J5.49	P9.143	TST_HDRA108	U27.AB33
J1.144	P3N59	J6.1	P9.144	TST_HDRA109	U27.AB36
J1.145	P3N58	J6.3	P9.145	TST_HDRA110	U27.AB37
J1.146	P3N53	J6.5	P9.146	TST_HDRA111	U27.AB40
J1.147	P3N52	J6.7	P9.147	TST_HDRA112	U27.AA31
J1.148	P3N51	J2.17	P9.148	TST_HDRA113	U27.AA34
J1.149	P3N50	J2.18	P9.149	TST_HDRA114	U27.Y31
J1.150	P3N45	J6.9	P9.150	TST_HDRA115	U27.Y33
J1.151	No Connect		P9.151	GND	
J1.152	P3N44	J6.11	P9.152	TST_HDRA116	U27.Y37

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.153	P3N41	J6.13	P9.153	TST_HDRA117	U27.W32
J1.154	P3N40	J6.15	P9.154	TST_HDRA118	U27.W34
J1.155	P3N37	J6.17	P9.155	TST_HDRA119	U27.W36
J1.156	P3N36	J6.19	P9.156	TST_HDRA120	U27.W37
J1.157	P3N33	J6.21	P9.157	TST_HDRA121	U27.W42
J1.158	P3N32	J6.23	P9.158	TST_HDRA122	U27.V32
J1.159	P3N31	J2.44	P9.159	TST_HDRA123	U27.V35
J1.160	P3N30	J2.45	P9.160	TST_HDRA124	U27.V41
J1.161	P3N25	J6.25	P9.161	TST_HDRA125	U27.U31
J1.162	No Connect		P9.162	GND	
J1.163	P3N24	J6.27	P9.163	TST_HDRA126	U27.U34
J1.164	P3N21	J6.29	P9.164	TST_HDRA127	U27.U36
J1.165	P3N20	J6.31	P9.165	TST_HDRA128	U27.U41
J1.166	P3N17	J6.33	P9.166	TST_HDRA129	U27.T31
J1.167	P3N16	J6.35	P9.167	TST_HDRA130	U27.T32
J1.168	P3N13	J6.37	P9.168	TST_HDRA131	U27.T35
J1.169	P3N12	J6.39	P9.169	TST_HDRA132	U27.T41
J1.170	P3N11	J2.47	P9.170	TST_HDRA133	U27.R32
J1.171	P3N10	J2.48	P9.171	TST_HDRA134	U27.R34
J1.172	P3N5	J6.41	P9.172	TST_HDRA135	U27.R41
J1.173	No Connect		P9.173	GND	
J1.174	P3N4	J6.43	P9.174	TST_HDRA136	U27.P32
J1.175	P3N1	J6.45	P9.175	TST_HDRA137	U27.P34
J1.176	P3N0	J6.47	P9.176	TST_HDRA138	U27.P36
J1.177	P4N25	J7.1	P9.177	TST_HDRA139	U27.P42
J1.178	P4N24	J7.3	P9.178	TST_HDRA140	U27.N31

Daughter Card Connections			DN6000K10 IO Connections Test Header A		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.179	P4N23	J7.5	P9.179	TST_HDRA141	U27.N33
J1.180	P4N22	J7.7	P9.180	TST_HDRA142	U27.N35
J1.181	P4N17	J7.9	P9.181	TST_HDRA143	U27.N41
J1.182	P4N16	J7.11	P9.182	TST_HDRA144	U27.M31
J1.183	P4N15	J7.13	P9.183	TST_HDRA145	U27.M33
J1.184	GND	J2.36	P9.184	GND	
J1.185	P4N14	J7.15	P9.185	TST_HDRA146	U27.M36
J1.186	P4N9	J7.17	P9.186	TST_HDRA147	U27.L33
J1.187	P4N8	J7.19	P9.187	TST_HDRA148	U27.L35
J1.188	P4N5	J7.21	P9.188	TST_HDRA149	U27.L41
J1.189	P4N4	J7.23	P9.189	TST_HDRA150	U27.L42
J1.190	P4N1	J7.25	P9.190	TST_HDRA151	U27.K35
J1.191	P4N0	J7.27	P9.191	TST_HDRA152	U27.K41
J1.192	P4NX13	J7.29	P9.192	TST_HDRA153	U27.J35
J1.193	P4NX12	J7.31	P9.193	TST_HDRA154	U27.J41
J1.194	P4NX9	J7.33	P9.194	TST_HDRA155	U27.J42
J1.195	No Connect		P9.195	GND	
J1.196	P4NX8	J7.35	P9.196	TST_HDRA156	U27.H36
J1.197	P4NX3	J7.37	P9.197	TST_HDRA157	U27.G41
J1.198	P4NX2	J7.39	P9.198	TST_HDRA158	U27.F41
J1.199	P4NX1	J7.41	P9.199	TST_HDRA159	U27.E41
J1.200	P4NX0	J7.43	P9.200	TST_HDRA160	U27.D41

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.001	No Connect		P10.1	+12V	
J1.002	No Connect		P10.2	GND	
J1.003	ACLK1	J5.1	P10.3	+2.5V	
J1.004	No Connect		P10.4	+5V	
J1.005	BCLK1	J5.3	P10.5	+2.5V	
J1.006	No Connect		P10.6	+5V	
J1.007	CCLK1	J5.5	P10.7	ACLK10	
J1.008	No Connect		P10.8	GND	
J1.009	No Connect		P10.9	+3.3V	
J1.010	BP2N3(P2N3)	J3.1	P10.10	BCLK10	
J1.011	No Connect		P10.11	GND	
J1.012	BP2N2(P2N2)	J3.3	P10.12	FD2	U53.D2 and U51.AW2
J1.013	P2N1	J2.8	P10.13	FD4	U53.E1 and U51.AV1
J1.014	P2N0	J2.9	P10.14	FD6	U53.E3 and U51.AV3
J1.015	BP2NX7(P2NX7)	J3.5	P10.15	FD8	U53.F2 and U51.AU2
J1.016	BP2NX6(P2NX6)	J3.7	P10.16	FD10	U53.G1 and U51.AT1
J1.017	BP2NX5(P2NX5)	J3.9	P10.17	FD12	U53.G3 and U51.AT3
J1.018	BP2NX4(P2NX4)	J3.11	P10.18	FD14	U53.H3 and U51.AR3
J1.019	P2NX1	J2.10	P10.19	FD16	U53.J2 and U51.AP2
J1.020	P2NX0	J2.11	P10.20	FD18	U53.K2 and U51.AN2

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.021	P3NX9	J2.40	P10.21	FD20	U53.L1 and U51.AM1
J1.022	No Connect		P10.22	GND	
J1.023	P3NX8	J2.41	P10.23	FD22	U53.L3 and U51.AM3
J1.024	BP3NX5(P3NX5)	J3.13	P10.24	FD24	U53.M3 and U51.AL3
J1.025	BP3NX4(P3NX4)	J3.15	P10.25	FD26	U53.N2 and U51.AK2
J1.026	BP3N89(P3N89)	J3.17	P10.26	FD28	U53.P1 and U51.AJ1
J1.027	BP3N88(P3N88)	J3.19	P10.27	FD30	U53.P3 and U51.AJ3
J1.028	BP3N87(P3N87)	J3.21	P10.28	FD32	U53.R2 and U51.AH2
J1.029	BP3N86(P3N86)	J3.23	P10.29	FD34	U53.T2 and U51.AG2
J1.030	BP3N83(P3N83)	J3.25	P10.30	FD35	U53.T3 and U51.AG3
J1.031	BP3N82(P3N82)	J3.27	P10.31	FD37	U53.U2 and U51.AF2
J1.032	BP3N77(P3N77)	J3.29	P10.32	FD39	U53.V1 and U51.AE1
J1.033	No Connect		P10.33	GND	
J1.034	BP3N76(P3N76)	J3.31	P10.34	FD42	U53.W2 and U51.AD2
J1.035	BP3N75(P3N75)	J3.33	P10.35	FD44	U53.W4 and U51.AD4
J1.036	BP3N74(P3N74)	J3.35	P10.36	FD46	U53.Y4 and U51.AC4
J1.037	P3N69	J2.42	P10.37	FD48	U53.AA4

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
					and U51.AB4
J1.038	P3N68	J2.43	P10.38	TST_HDRB24	U52.D2
J1.039	BP3N67(P3N67)	J3.37	P10.39	TST_HDRB25	U52.E2
J1.040	BP3N66(P3N66)	J3.39	P10.40	TST_HDRB26	U52.F2
J1.041	BP3N63(P3N63)	J3.41	P10.41	TST_HDRB27	U52.G2
J1.042	BP3N62(P3N62)	J3.43	P10.42	TST_HDRB28	U52.J1
J1.043	BP3N57(P3N57)	J3.45	P10.43	TST_HDRB29	U52.K1
J1.044	No Connect		P10.44	GND	
J1.045	BP3N56(P3N56)	J3.47	P10.45	TST_HDRB30	U52.L1
J1.046	No Connect		P10.46	TST_HDRB31	U52.M2
J1.047	No Connect		P10.47	TST_HDRB32	U52.N2
J1.048	BP3N49(P3N49)	J4.1	P10.48	TST_HDRB33	U52.P2
J1.049	BP3N48(P3N48)	J4.3	P10.49	TST_HDRB34	U52.R2
J1.050	P3N47	J2.19	P10.50	TST_HDRB35	U52.U1
J1.051	P3N46	J2.20	P10.51	TST_HDRB36	U52.V1
J1.052	BP3N43(P3N43)	J4.5	P10.52	TST_HDRB37	U52.W1
J1.053	BP3N42(P3N42)	J4.7	P10.53	TST_HDRB38	U52.W3
J1.054	BP3N39(P3N39)	J4.9	P10.54	TST_HDRB39	U52.AA3
J1.055	No Connect		P10.55	GND	
J1.056	BP3N38(P3N38)	J4.11	P10.56	TST_HDRB40	U52.AC3
J1.057	BP3N35(P3N35)	J4.13	P10.57	TST_HDRB41	U52.AD2
J1.058	BP3N34(P3N34)	J4.15	P10.58	TST_HDRB42	U52.AE2
J1.059	BP3N29(P3N29)	J4.17	P10.59	TST_HDRB43	U52.AF2
J1.060	BP3N28(P3N28)	J4.19	P10.60	TST_HDRB44	U52.AH1
J1.061	BP3N27(P3N27)	J4.21	P10.61	TST_HDRB45	U52.AJ1
J1.062	BP3N26(P3N26)	J4.23	P10.62	TST_HDRB46	U52.AK1

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.063	P3N23	J2.21	P10.63	TST_HDRB47	U52.AL2
J1.064	P3N22	J2.22	P10.64	TST_HDRB48	U52.AM2
J1.065	BP3N19(P3N19)	J4.25	P10.65	TST_HDRB49	U52.AN2
J1.066	No Connect		P10.66	GND	
J1.067	BP3N18(P3N18)	J4.27	P10.67	TST_HDRB50	U52.AP2
J1.068	BP3N15(P3N15)	J4.29	P10.68	TST_HDRB51	U52.AT1
J1.069	BP3N14(P3N14)	J4.31	P10.69	TST_HDRB52	U52.AU1
J1.070	P3N9	J2.23	P10.70	TST_HDRB53	U52.AV1
J1.071	P3N8	J2.24	P10.71	TST_HDRB54	U52.AW1
J1.072	BP3N7(P3N7)	J4.33	P10.72	FD50	U53.AB4 and U51.AA4
J1.073	BP3N6(P3N6)	J4.35	P10.73	FD51	U53.AC3 and U51.Y3
J1.074	BP3N3(P3N3)	J4.37	P10.74	FD52	U53.AC4 and U51.Y4
J1.075	BP3N2(P3N2)	J4.39	P10.75	FD53	U53.AD1 and U51.W1
J1.076	BP4N27(P4N27)	J4.41	P10.76	FD55	U53.AD3 and U51.W3
J1.077	No Connect		P10.77	GND	
J1.078	BP4N26(P4N26)	J4.43	P10.78	FD58	U53.AE2 and U51.V2
J1.079	BP4N21(P4N21)	J4.45	P10.79	FD60	U53.AF2 and U51.U2
J1.080	BP4N20(P4N20)	J4.47	P10.80	FD62	U53.AG2 and U51.T2
J1.081	No Connect		P10.81	FD64	U53.AH1 and U51.R1
J1.082	No Connect		P10.82	FD66	U53.AH3

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
					and U51.R3
J1.083	No Connect		P10.83	FD68	U53.AJ2 and U51.P2
J1.084	No Connect		P10.84	FD70	U53.AK1 and U51.N1
J1.085	No Connect		P10.85	FD71	U53.AK2 and U51.N2
J1.086	No Connect		P10.86	FD73	U53.AL2 and U51.M2
J1.087	No Connect		P10.87	FD75	U53.AM1 and U51.L1
J1.088	No Connect		P10.88	GND	
J1.089	No Connect		P10.89	FD78	U53.AN1 and U51.K1
J1.090	No Connect		P10.90	FD80	U53.AN3 and U51.K3
J1.091	No Connect		P10.91	FD82	U53.AP2 and U51.J2
J1.092	No Connect		P10.92	FD84	U53.AR3 and U51.H3
J1.093	No Connect		P10.93	+1.5V	
J1.094	No Connect		P10.94	FD87	U53.AT3 and U51.G3
J1.095	P4NX7	J7.45	P10.95	FD89	U53.AU2 and U51.F2
J1.096	P4NX6	J7.47	P10.96	FD90	U53.AU3 and U51.F3
J1.097	No Connect		P10.97	FD92	U53.AV2 and U51.E2
J1.098	No Connect		P10.98	FD94	U53.AW1 and U51.D1

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.099	No Connect		P10.99	GND	
J1.100	No Connect		P10.100	-12V	
J1.101	No Connect		P10.101	GND	
J1.102	MBCK1	J2.27	P10.102	TST_HDRB_CLKIN	U51.AN22
J1.103	No Connect		P10.103	+1.5V	
J1.104	MBCK0	J2.28	P10.104	GND	
J1.105	No Connect		P10.105	+3.3V	
J1.107	No Connect		P10.107	GND	
J1.108	ECLK1	J5.7	P10.108	GND	
J1.109	No Connect		P10.109	GND	
J1.110	No Connect		P10.110	GND	
J1.111	P2N5	J5.15	P10.111	FD1	U53.D1 and U51.AW1
J1.112	P2N4	J5.17	P10.112	FD3	U53.D3 and U51.AW3
J1.113	P2NX11	J2.2	P10.113	FD5	U53.E2 and U51.AV2
J1.114	P2NX10	J2.1	P10.114	FD7	U53.F1 and U51.AU1
J1.115	P2NX9	J5.19	P10.115	FD9	U53.F3 and U51.AU3
J1.116	P2NX8	J5.21	P10.116	FD11	U53.G2 and U51.AT2
J1.117	P2NX3	J5.23	P10.117	FD13	U53.H2 and U51.AR2
J1.118	No Connect		P10.118	GND	
J1.119	P2NX2	J5.25	P10.119	FD15	U53.J1 and U51.AP1

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.120	P3NX11	J2.29	P10.120	FD17	U53.K1 and U51.AN1
J1.121	P3NX10	J2.30	P10.121	FD19	U53.K3 and U51.AN3
J1.122	P3NX7	J2.31	P10.122	FD21	U53.L2 and U51.AM2
J1.123	P3NX6	J2.32	P10.123	FD23	U53.M2 and U51.AL2
J1.124	P3NX3	J5.27	P10.124	FD25	U53.N1 and U51.AK1
J1.125	P3NX2	J5.29	P10.125	FD27	U53.N3 and U51.AK3
J1.126	P3NX1	J5.31	P10.126	FD29	U53.P2 and U51.AJ2
J1.127	P3NX0	J5.33	P10.127	FD31	U53.R1 and U51.AH1
J1.128	P3N85	J5.35	P10.128	FD33	U53.R3 and U51.AH3
J1.129	No Connect		P10.129	GND	
J1.130	P3N84	J5.37	P10.130	FD36	U53.U1 and U51.AF1
J1.131	P3N81	J5.39	P10.131	FD38	U53.U3 and U51.AF3
J1.132	P3N80	J5.41	P10.132	FD40	U53.V2 and U51.AE2
J1.133	P3N79	J2.3	P10.133	FD41	U53.W1 and U51.AD1
J1.134	P3N78	J2.4	P10.134	FD43	U53.W3 and U51.AD3
J1.135	P3N73	J2.6	P10.135	FD45	U53.Y3 and U51.AC3

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.136	P3N72	J2.7	P10.136	FD47	U53.AA3 and U51.AB3
J1.137	P3N71	J2.33	P10.137	FD49	U53.AB3 and U51.AA3
J1.138	P3N70	J2.34	P10.138	TST_HDRB104	U52.D1
J1.139	P3N65	J5.43	P10.139	TST_HDRB105	U52.E1
J1.140	No Connect		P10.140	GND	
J1.141	P3N64	J5.45	P10.141	TST_HDRB106	U52.F1
J1.142	P3N61	J5.47	P10.142	TST_HDRB107	U52.G1
J1.143	P3N60	J5.49	P10.143	TST_HDRB108	U52.H2
J1.144	P3N59	J6.1	P10.144	TST_HDRB109	U52.J2
J1.145	P3N58	J6.3	P10.145	TST_HDRB110	U52.K2
J1.146	P3N53	J6.5	P10.146	TST_HDRB111	U52.L2
J1.147	P3N52	J6.7	P10.147	TST_HDRB112	U52.N1
J1.148	P3N51	J2.17	P10.148	TST_HDRB113	U52.P1
J1.149	P3N50	J2.18	P10.149	TST_HDRB114	U52.R1
J1.150	P3N45	J6.9	P10.150	TST_HDRB115	U52.T2
J1.151	No Connect		P10.151	GND	
J1.152	P3N44	J6.11	P10.152	TST_HDRB116	U52.U2
J1.153	P3N41	J6.13	P10.153	TST_HDRB117	U52.V2
J1.154	P3N40	J6.15	P10.154	TST_HDRB118	U52.W2
J1.155	P3N37	J6.17	P10.155	TST_HDRB119	U52.Y3
J1.156	P3N36	J6.19	P10.156	TST_HDRB120	U52.AB3
J1.157	P3N33	J6.21	P10.157	TST_HDRB121	U52.AD3
J1.158	P3N32	J6.23	P10.158	TST_HDRB122	U52.AD1
J1.159	P3N31	J2.44	P10.159	TST_HDRB123	U52.AE1

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.160	P3N30	J2.45	P10.160	TST_HDRB124	U52.AF1
J1.161	P3N25	J6.25	P10.161	TST_HDRB125	U52.AG2
J1.162	No Connect		P10.162	GND	
J1.163	P3N24	J6.27	P10.163	TST_HDRB126	U52.AH2
J1.164	P3N21	J6.29	P10.164	TST_HDRB127	U52.AJ2
J1.165	P3N20	J6.31	P10.165	TST_HDRB128	U52.AK2
J1.166	P3N17	J6.33	P10.166	TST_HDRB129	U52.AM1
J1.167	P3N16	J6.35	P10.167	TST_HDRB130	U52.AN1
J1.168	P3N13	J6.37	P10.168	TST_HDRB131	U52.AP1
J1.169	P3N12	J6.39	P10.169	TST_HDRB132	U52.AR2
J1.170	P3N11	J2.47	P10.170	TST_HDRB133	U52.AT2
J1.171	P3N10	J2.48	P10.171	TST_HDRB134	U52.AU2
J1.172	P3N5	J6.41	P10.172	TST_HDRB135	U52.AV2
J1.173	No Connect		P10.173	GND	
J1.174	P3N4	J6.43	P10.174	TST_HDRB136	U52.AW2
J1.175	P3N1	J6.45	P10.175	FD54	U53.AD2 and U51.W2
J1.176	P3N0	J6.47	P10.176	FD56	U53.AD4 and U51.W4
J1.177	P4N25	J7.1	P10.177	FD57	U53.AE1 and U51.V1
J1.178	P4N24	J7.3	P10.178	FD59	U53.AF1 and U51.U1
J1.179	P4N23	J7.5	P10.179	FD61	U53.AF3 and U51.U3
J1.180	P4N22	J7.7	P10.180	FD63	U53.AG3 and U51.T3
J1.181	P4N17	J7.9	P10.181	FD65	U53.AH2 and U51.R2

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.182	P4N16	J7.11	P10.182	FD67	U53.AJ1 and U51.P1
J1.183	P4N15	J7.13	P10.183	FD69	U53.AJ3 and U51.P3
J1.184	GND	J2.36	P10.184	GND	
J1.185	P4N14	J7.15	P10.185	FD72	U53.AK3 and U51.N3
J1.186	P4N9	J7.17	P10.186	FD74	U53.AL3 and U51.M3
J1.187	P4N8	J7.19	P10.187	FD76	U53.AM2 and U51.L2
J1.188	P4N5	J7.21	P10.188	FD77	U53.AM3 and U51.L3
J1.189	P4N4	J7.23	P10.189	FD79	U53.AN2 and U51.K2
J1.190	P4N1	J7.25	P10.190	FD81	U53.AP1 and U51.J1
J1.191	P4N0	J7.27	P10.191	FD83	U53.AR2 and U51.H2
J1.192	P4NX13	J7.29	P10.192	FD85	U53.AT1 and U51.G1
J1.193	P4NX12	J7.31	P10.193	FD86	U53.AT2 and U51.G2
J1.194	P4NX9	J7.33	P10.194	FD88	U53.AU1 and U51.F1
J1.195	No Connect		P10.195	GND	
J1.196	P4NX8	J7.35	P10.196	FD91	U53.AV1 and U51.E1
J1.197	P4NX3	J7.37	P10.197	FD93	U53.AV3 and U51.E3
J1.198	P4NX2	J7.39	P10.198	FD95	U53.AW2

Daughter Card Connections			DN6000K10 IO Connections Test Header B		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
					and U51.D2
J1.199	P4NX1	J7.41	P10.199	FD96	U53.AW3 and U51.D3

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.001	No Connect		P11.1	+12V	
J1.002	No Connect		P11.2	GND	
J1.003	ACLK1	J5.1	P11.3	+2.5V	
J1.004	No Connect		P11.4	+5V	
J1.005	BCLK1	J5.3	P11.5	+2.5V	
J1.006	No Connect		P11.6	+5V	
J1.007	CCLK1	J5.5	P11.7	ACLK11	
J1.008	No Connect		P11.8	GND	
J1.009	No Connect		P11.9	+3.3V	
J1.010	BP2N3(P2N3)	J3.1	P11.10	BCLK11	
J1.011	No Connect		P11.11	GND	
J1.012	BP2N2(P2N2)	J3.3	P11.12	TST_HDRC0	U80.AD1
J1.013	P2N1	J2.8	P11.13	TST_HDRC1	U80.AE1
J1.014	P2N0	J2.9	P11.14	TST_HDRC2	U80.AF1
J1.015	BP2NX7(P2NX7)	J3.5	P11.15	TST_HDRC3	U80.AF3
J1.016	BP2NX6(P2NX6)	J3.7	P11.16	TST_HDRC4	U80.AG2
J1.017	BP2NX5(P2NX5)	J3.9	P11.17	TST_HDRC5	U80.AH1
J1.018	BP2NX4(P2NX4)	J3.11	P11.18	TST_HDRC6	U80.AH3
J1.019	P2NX1	J2.10	P11.19	TST_HDRC7	U80.AJ2
J1.020	P2NX0	J2.11	P11.20	TST_HDRC8	U80.AK1
J1.021	P3NX9	J2.40	P11.21	TST_HDRC9	U80.AK3
J1.022	No Connect		P11.22	GND	
J1.023	P3NX8	J2.41	P11.23	TST_HDRC10	U80.AL3
J1.024	BP3NX5(P3NX5)	J3.13	P11.24	TST_HDRC11	U80.AM2
J1.025	BP3NX4(P3NX4)	J3.15	P11.25	TST_HDRC12	U80.AN1
J1.026	BP3N89(P3N89)	J3.17	P11.26	TST_HDRC13	U80.AN3

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.027	BP3N88(P3N88)	J3.19	P11.27	TST_HDRC14	U80.AP2
J1.028	BP3N87(P3N87)	J3.21	P11.28	TST_HDRC15	U80.AR3
J1.029	BP3N86(P3N86)	J3.23	P11.29	TST_HDRC16	U80.AT2
J1.030	BP3N83(P3N83)	J3.25	P11.30	TST_HDRC17	U80.AU1
J1.031	BP3N82(P3N82)	J3.27	P11.31	TST_HDRC18	U80.AU3
J1.032	BP3N77(P3N77)	J3.29	P11.32	TST_HDRC19	U80.AV2
J1.033	No Connect		P11.33	GND	
J1.034	BP3N76(P3N76)	J3.31	P11.34	TST_HDRC20	U80.AW1
J1.035	BP3N75(P3N75)	J3.33	P11.35	TST_HDRC21	U78.D2
J1.036	BP3N74(P3N74)	J3.35	P11.36	TST_HDRC22	U78.E1
J1.037	P3N69	J2.42	P11.37	TST_HDRC23	U78.E3
J1.038	P3N68	J2.43	P11.38	TST_HDRC24	U78.F2
J1.039	BP3N67(P3N67)	J3.37	P11.39	TST_HDRC25	U78.G1
J1.040	BP3N66(P3N66)	J3.39	P11.40	TST_HDRC26	U78.G3
J1.041	BP3N63(P3N63)	J3.41	P11.41	TST_HDRC27	U78.H3
J1.042	BP3N62(P3N62)	J3.43	P11.42	TST_HDRC28	U78.J2
J1.043	BP3N57(P3N57)	J3.45	P11.43	TST_HDRC29	U78.K2
J1.044	No Connect		P11.44	GND	
J1.045	BP3N56(P3N56)	J3.47	P11.45	TST_HDRC30	U78.L1
J1.046	No Connect		P11.46	TST_HDRC31	U78.L3
J1.047	No Connect		P11.47	TST_HDRC32	U78.M3
J1.048	BP3N49(P3N49)	J4.1	P11.48	TST_HDRC33	U78.N2
J1.049	BP3N48(P3N48)	J4.3	P11.49	TST_HDRC34	U78.P1
J1.050	P3N47	J2.19	P11.50	TST_HDRC35	U78.R1
J1.051	P3N46	J2.20	P11.51	TST_HDRC36	U78.T2
J1.052	BP3N43(P3N43)	J4.5	P11.52	TST_HDRC37	U78.U2

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.053	BP3N42(P3N42)	J4.7	P11.53	TST_HDRC38	U78.V2
J1.054	BP3N39(P3N39)	J4.9	P11.54	TST_HDRC39	U78.AD1
J1.055	No Connect		P11.55	GND	
J1.056	BP3N38(P3N38)	J4.11	P11.56	TST_HDRC40	U78.AE1
J1.057	BP3N35(P3N35)	J4.13	P11.57	TST_HDRC41	U78.AF1
J1.058	BP3N34(P3N34)	J4.15	P11.58	TST_HDRC42	U78.AG2
J1.059	BP3N29(P3N29)	J4.17	P11.59	TST_HDRC43	U78.AH2
J1.060	BP3N28(P3N28)	J4.19	P11.60	TST_HDRC44	U78.AJ2
J1.061	BP3N27(P3N27)	J4.21	P11.61	TST_HDRC45	U78.AK2
J1.062	BP3N26(P3N26)	J4.23	P11.62	TST_HDRC46	U78.AL2
J1.063	P3N23	J2.21	P11.63	TST_HDRC47	U78.AM1
J1.064	P3N22	J2.22	P11.64	TST_HDRC48	U78.AM3
J1.065	BP3N19(P3N19)	J4.25	P11.65	TST_HDRC49	U78.AN2
J1.066	No Connect		P11.66	GND	
J1.067	BP3N18(P3N18)	J4.27	P11.67	TST_HDRC50	U78.AP1
J1.068	BP3N15(P3N15)	J4.29	P11.68	TST_HDRC51	U78.AR2
J1.069	BP3N14(P3N14)	J4.31	P11.69	TST_HDRC52	U78.AT1
J1.070	P3N9	J2.23	P11.70	TST_HDRC53	U78.AT3
J1.071	P3N8	J2.24	P11.71	TST_HDRC54	U78.AU2
J1.072	BP3N7(P3N7)	J4.33	P11.72	TST_HDRC55	U78.AV1
J1.073	BP3N6(P3N6)	J4.35	P11.73	TST_HDRC56	U78.AV3
J1.074	BP3N3(P3N3)	J4.37	P11.74	TST_HDRC57	U78.AW2
J1.075	BP3N2(P3N2)	J4.39	P11.75	TST_HDRC58	U79.D1
J1.076	BP4N27(P4N27)	J4.41	P11.76	TST_HDRC59	U79.D3
J1.077	No Connect		P11.77	GND	
J1.078	BP4N26(P4N26)	J4.43	P11.78	TST_HDRC60	U79.E2

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.079	BP4N21(P4N21)	J4.45	P11.79	TST_HDRC61	U79.F1
J1.080	BP4N20(P4N20)	J4.47	P11.80	TST_HDRC62	U79.F3
J1.081	No Connect		P11.81	TST_HDRC63	U79.G2
J1.082	No Connect		P11.82	TST_HDRC64	U79.H2
J1.083	No Connect		P11.83	TST_HDRC65	U79.J1
J1.084	No Connect		P11.84	TST_HDRC66	U79.K1
J1.085	No Connect		P11.85	TST_HDRC67	U79.K3
J1.086	No Connect		P11.86	TST_HDRC68	U79.L2
J1.087	No Connect		P11.87	TST_HDRC69	U79.M2
J1.088	No Connect		P11.88	GND	
J1.089	No Connect		P11.89	TST_HDRC70	U79.N1
J1.090	No Connect		P11.90	TST_HDRC71	U79.N3
J1.091	No Connect		P11.91	TST_HDRC72	U79.P2
J1.092	No Connect		P11.92	TST_HDRC73	U79.R1
J1.093	No Connect		P11.93	+1.5V	
J1.094	No Connect		P11.94	TST_HDRC74	U79.R3
J1.095	P4NX7	J7.45	P11.95	TST_HDRC75	U79.T3
J1.096	P4NX6	J7.47	P11.96	TST_HDRC76	U79.U2
J1.097	No Connect		P11.97	TST_HDRC77	U79.V2
J1.098	No Connect		P11.98	TST_HDRC78	U79.W2
J1.099	No Connect		P11.99	GND	
J1.100	No Connect		P11.100	-12V	
J1.101	No Connect		P11.101	GND	
J1.102	MBCK1	J2.27	P11.102	TST_HDRC_CLKIN	U80.AN22
J1.103	No Connect		P11.103	+1.5V	
J1.104	MBCK0	J2.28	P11.104	GND	

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.105	No Connect		P11.105	+3.3V	
J1.107	No Connect		P11.107	GND	
J1.108	ECLK1	J5.7	P11.108	GND	
J1.109	No Connect		P11.109	GND	
J1.110	No Connect		P11.110	GND	
J1.111	P2N5	J5.15	P11.111	TST_HDRC79	U80.AD2
J1.112	P2N4	J5.17	P11.112	TST_HDRC80	U80.AE2
J1.113	P2NX11	J2.2	P11.113	TST_HDRC81	U80.AF2
J1.114	P2NX10	J2.1	P11.114	TST_HDRC82	U80.AG3
J1.115	P2NX9	J5.19	P11.115	TST_HDRC83	U80.AH2
J1.116	P2NX8	J5.21	P11.116	TST_HDRC84	U80.AJ1
J1.117	P2NX3	J5.23	P11.117	TST_HDRC85	U80.AJ3
J1.118	No Connect		P11.118	GND	
J1.119	P2NX2	J5.25	P11.119	TST_HDRC86	U80.AK2
J1.120	P3NX11	J2.29	P11.120	TST_HDRC87	U80.AL2
J1.121	P3NX10	J2.30	P11.121	TST_HDRC88	U80.AM1
J1.122	P3NX7	J2.31	P11.122	TST_HDRC89	U80.AM3
J1.123	P3NX6	J2.32	P11.123	TST_HDRC90	U80.AN2
J1.124	P3NX3	J5.27	P11.124	TST_HDRC91	U80.AP1
J1.125	P3NX2	J5.29	P11.125	TST_HDRC92	U80.AR2
J1.126	P3NX1	J5.31	P11.126	TST_HDRC93	U80.AT1
J1.127	P3NX0	J5.33	P11.127	TST_HDRC94	U80.AT3
J1.128	P3N85	J5.35	P11.128	TST_HDRC95	U80.AU2
J1.129	No Connect		P11.129	GND	
J1.130	P3N84	J5.37	P11.130	TST_HDRC96	U80.AV1
J1.131	P3N81	J5.39	P11.131	TST_HDRC97	U80.AV3

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.132	P3N80	J5.41	P11.132	TST_HDRC98	U80.AW2
J1.133	P3N79	J2.3	P11.133	TST_HDRC99	U80.AW3
J1.134	P3N78	J2.4	P11.134	TST_HDRC100	U78.D1
J1.135	P3N73	J2.6	P11.135	TST_HDRC101	U78.D3
J1.136	P3N72	J2.7	P11.136	TST_HDRC102	U78.E2
J1.137	P3N71	J2.33	P11.137	TST_HDRC103	U78.F1
J1.138	P3N70	J2.34	P11.138	TST_HDRC104	U78.F3
J1.139	P3N65	J5.43	P11.139	TST_HDRC105	U78.G2
J1.140	No Connect		P11.140	GND	
J1.141	P3N64	J5.45	P11.141	TST_HDRC106	U78.H2
J1.142	P3N61	J5.47	P11.142	TST_HDRC107	U78.J1
J1.143	P3N60	J5.49	P11.143	TST_HDRC108	U78.K1
J1.144	P3N59	J6.1	P11.144	TST_HDRC109	U78.K3
J1.145	P3N58	J6.3	P11.145	TST_HDRC110	U78.L2
J1.146	P3N53	J6.5	P11.146	TST_HDRC111	U78.M2
J1.147	P3N52	J6.7	P11.147	TST_HDRC112	U78.N1
J1.148	P3N51	J2.17	P11.148	TST_HDRC113	U78.N3
J1.149	P3N50	J2.18	P11.149	TST_HDRC114	U78.P2
J1.150	P3N45	J6.9	P11.150	TST_HDRC115	U78.R2
J1.151	No Connect		P11.151	GND	
J1.152	P3N44	J6.11	P11.152	TST_HDRC116	U78.U1
J1.153	P3N41	J6.13	P11.153	TST_HDRC117	U78.V1
J1.154	P3N40	J6.15	P11.154	TST_HDRC118	U78.W1
J1.155	P3N37	J6.17	P11.155	TST_HDRC119	U78.W2
J1.156	P3N36	J6.19	P11.156	TST_HDRC120	U78.Y9
J1.157	P3N33	J6.21	P11.157	TST_HDRC121	U78.AD2

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.158	P3N32	J6.23	P11.158	TST_HDRC122	U78.AE2
J1.159	P3N31	J2.44	P11.159	TST_HDRC123	U78.AF2
J1.160	P3N30	J2.45	P11.160	TST_HDRC124	U78.AH1
J1.161	P3N25	J6.25	P11.161	TST_HDRC125	U78.AJ1
J1.162	No Connect		P11.162	GND	
J1.163	P3N24	J6.27	P11.163	TST_HDRC126	U78.AK1
J1.164	P3N21	J6.29	P11.164	TST_HDRC127	U78.AK3
J1.165	P3N20	J6.31	P11.165	TST_HDRC128	U78.AL3
J1.166	P3N17	J6.33	P11.166	TST_HDRC129	U78.AM2
J1.167	P3N16	J6.35	P11.167	TST_HDRC130	U78.AN1
J1.168	P3N13	J6.37	P11.168	TST_HDRC131	U78.AN3
J1.169	P3N12	J6.39	P11.169	TST_HDRC132	U78.AP2
J1.170	P3N11	J2.47	P11.170	TST_HDRC133	U78.AR3
J1.171	P3N10	J2.48	P11.171	TST_HDRC134	U78.AT2
J1.172	P3N5	J6.41	P11.172	TST_HDRC135	U78.AU1
J1.173	No Connect		P11.173	GND	
J1.174	P3N4	J6.43	P11.174	TST_HDRC136	U78.AU3
J1.175	P3N1	J6.45	P11.175	TST_HDRC137	U78.AV2
J1.176	P3N0	J6.47	P11.176	TST_HDRC138	U78.AW1
J1.177	P4N25	J7.1	P11.177	TST_HDRC139	U78.AW3
J1.178	P4N24	J7.3	P11.178	TST_HDRC140	U79.D2
J1.179	P4N23	J7.5	P11.179	TST_HDRC141	U79.E1
J1.180	P4N22	J7.7	P11.180	TST_HDRC142	U79.E3
J1.181	P4N17	J7.9	P11.181	TST_HDRC143	U79.F2
J1.182	P4N16	J7.11	P11.182	TST_HDRC144	U79.G1
J1.183	P4N15	J7.13	P11.183	TST_HDRC145	U79.G3

Daughter Card Connections			DN6000K10 IO Connections Test Header C		
Test Header	Signal Name	Connector	Test Header	Signal Name	FPGA Pin
J1.184	GND	J2.36	P11.184	GND	
J1.185	P4N14	J7.15	P11.185	TST_HDRC146	U79.H3
J1.186	P4N9	J7.17	P11.186	TST_HDRC147	U79.J2
J1.187	P4N8	J7.19	P11.187	TST_HDRC148	U79.K2
J1.188	P4N5	J7.21	P11.188	TST_HDRC149	U79.L1
J1.189	P4N4	J7.23	P11.189	TST_HDRC150	U79.L3
J1.190	P4N1	J7.25	P11.190	TST_HDRC151	U79.M3
J1.191	P4N0	J7.27	P11.191	TST_HDRC152	U79.N2
J1.192	P4NX13	J7.29	P11.192	TST_HDRC153	U79.P1
J1.193	P4NX12	J7.31	P11.193	TST_HDRC154	U79.P3
J1.194	P4NX9	J7.33	P11.194	TST_HDRC155	U79.R2
J1.195	No Connect		P11.195	GND	
J1.196	P4NX8	J7.35	P11.196	TST_HDRC156	U79.T2
J1.197	P4NX3	J7.37	P11.197	TST_HDRC157	U79.U1
J1.198	P4NX2	J7.39	P11.198	TST_HDRC158	U79.U3
J1.199	P4NX1	J7.41	P11.199	TST_HDRC159	U79.V1
J1.200	P4NX0	J7.43	P11.200	TST_HDRC160	U79.W1

12 Mechanical

Two bus bars, MP1 and MP2 are installed to prevent flexing of the PWB. They are connected to the ground plane and can be used to ground test equipment. The user must not short any power rails or signals to these metal bars - they can conduct a lot of current. Mounting holes are provided to allow the PCB to be mounted in a case.

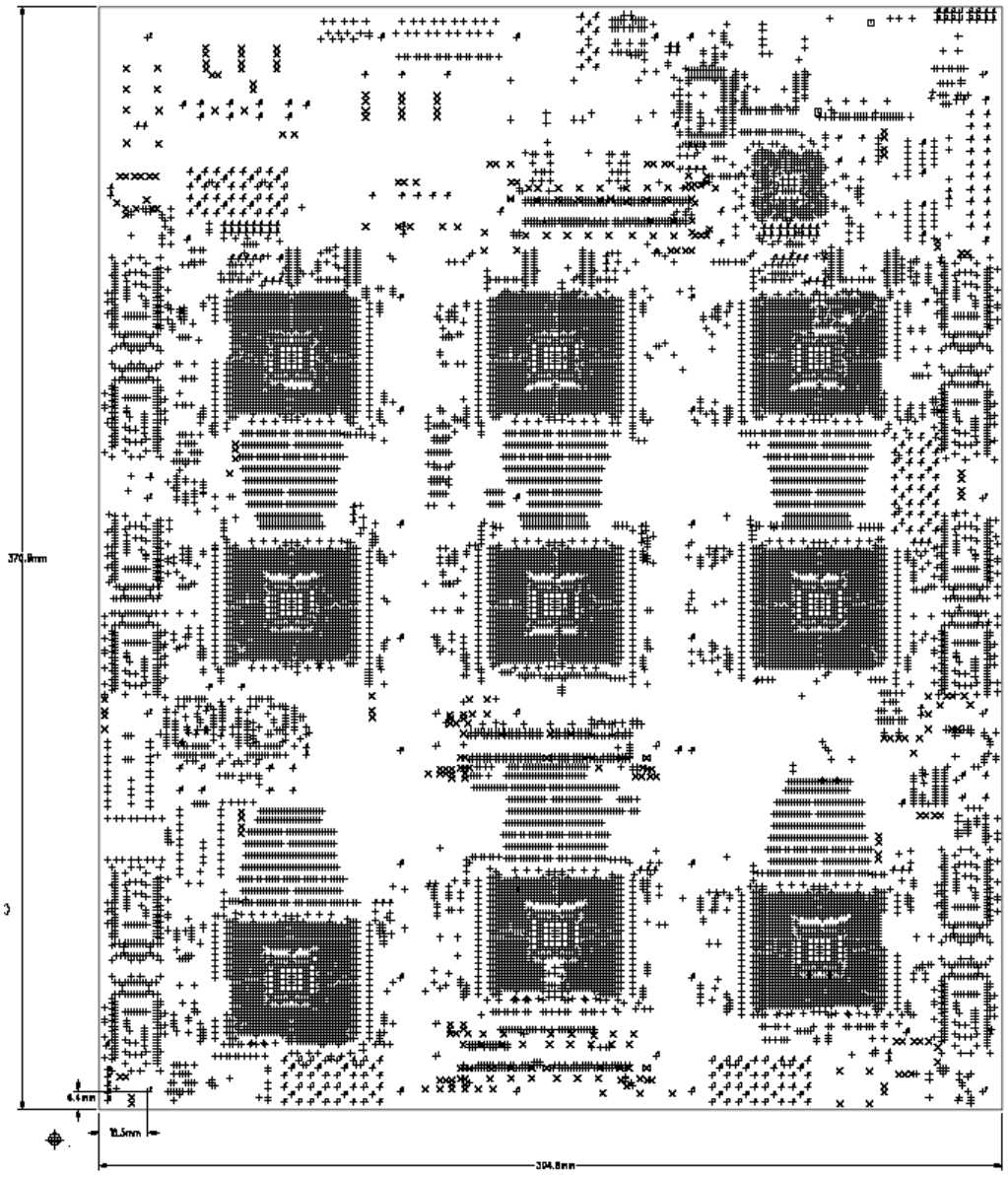
12.1.1 Case

The Amaquest PM7200 Server case holds the DN6000K10 board as well as a PC platform, refer to Figure 48.

Figure 48 - PM7200 Server Case

12.1.2 PWB Dimension

The DN6000K10 PWB conforms to the following dimensions:



Chapter 8

Appendix A – Address Maps

The DN6000k10 reference design can be used to verify the functionality of the board. There are several ways to exercise the reference design features. In each FPGA there is PowerPC code that allows the user to communicate directly with the FPGA through the RS232 port that is discussed in Using the Reference Design. Another method of communication is through USB (J3) or the Cypress MCU RS232 port P2. The USB PC application can be found on the product CD in "Source Code\USBController\USBController.exe". This application allows the user to read/write to different FPGA addresses and also perform tests on the DDR, FLASH, internal registers, and interconnect between the FPGA's (Description of Main Menu Options). The following 9 tables are the address maps for each FPGA when communicating through USB or via the RS232 port on the MCU (P2). Please note these address maps are not the same for communication through the PPC RS232 port menus (please see Using the Reference Design for a description). Also note that The Dini Group reference design provided with the DN6000k10 must be loaded in each of the existing FPGA's for the following address maps to be valid.

FPGA A

	Start Address	End Address	Read / Write	Description
DDR 1 (U32)	0x0000_0000	0x00FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U22)	0x0100_0000	0x01FF_FFFF	R/W	Address maps directly to DDR 2 (U22 only avail if 2vp100)
FLASH (U15)	0x0800_0000	0x08FF_FFFF	R/W	Address maps directly to FLASH
DDR Phase Shift Register	0x0C00_0000	0x0C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x0C00_0004	0x0C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x0C00_0008	0x0C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x0C00_0018	0x0C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA B

	Start Address	End Address	Read / Write	Description
FLASH (U16)	0x1800_0000	0x18FF_FFFF	R/W	Address maps directly to FLASH
External Host Commands Register	0x1C00_0004	0x1C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x1C00_0008	0x1C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x1C00_0018	0x1C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA C

	Start Address	End Address	Read / Write	Description
DDR 1 (U23)	0x2000_0000	0x20FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U34)	0x2100_0000	0x21FF_FFFF	R/W	Address maps directly to DDR 2 (U34 only avail if 2vp100)
FLASH (U17)	0x2800_0000	0x28FF_FFFF	R/W	Address maps directly to FLASH
DDR Phase Shift Register	0x2C00_0000	0x2C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x2C00_0004	0x2C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x2C00_0008	0x2C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x2C00_0018	0x2C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA D

	Start Address	End Address	Read / Write	Description
DDR 1 (U46)	0x3000_0000	0x30FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U58)	0x3100_0000	0x31FF_FFFF	R/W	Address maps directly to DDR 2 (U58 only avail if 2vp100)
FLASH (U15) FIXME?	0x3800_0000	0x38FF_FFFF	R/W	Address maps directly to FLASH
DDR Phase Shift Register	0x3C00_0000	0x3C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x3C00_0004	0x3C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x3C00_0008	0x3C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x3C00_0018	0x3C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA E

	Start Address	End Address	Read / Write	Description
External Host Commands Register	0x4C00_0004	0x4C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x4C00_0008	0x4C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x4C00_0018	0x4C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA F

	Start Address	End Address	Read / Write	Description
DDR 1 (U45)	0x5000_0000	0x50FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U57)	0x5100_0000	0x51FF_FFFF	R/W	Address maps directly to DDR 2 (U57 only avail if 2vp100)
DDR Phase Shift Register	0x5C00_0000	0x5C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x5C00_0004	0x5C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x5C00_0008	0x5C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x5C00_0018	0x5C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA G

	Start Address	End Address	Read / Write	Description
DDR 1 (U78)	0x6000_0000	0x60FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U85)	0x6100_0000	0x61FF_FFFF	R/W	Address maps directly to DDR 2 (U85 only avail if 2vp100)
FLASH (U88)	0x6800_0000	0x68FF_FFFF	R/W	Address maps directly to FLASH
DDR Phase Shift Register	0x6C00_0000	0x6C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x6C00_0004	0x6C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x6C00_0008	0x6C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x6C00_0018	0x6C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA H

	Start Address	End Address	Read / Write	Description
External Host Commands Register	0x7C00_0004	0x7C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x7C00_0008	0x7C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x7C00_0018	0x7C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70

FPGA I

	Start Address	End Address	Read / Write	Description
DDR 1 (U73)	0x8000_0000	0x80FF_FFFF	R/W	Address maps directly to DDR 1
DDR 2 (U83)	0x8100_0000	0x81FF_FFFF	R/W	Address maps directly to DDR 2 (U83 only avail if 2vp100)
FLASH (U86)	0x8800_0000	0x88FF_FFFF	R/W	Address maps directly to FLASH
DDR Phase Shift Register	0x8C00_0000	0x8C00_0000	R/W	DDR phase shift value (upper WORD is read only and contains the current phase shift value, lower WORD is write only)
External Host Commands Register	0x8C00_0004	0x8C00_0004	R/W	Write/Read register for MCU to issue the following commands: 0x1 – test all functionality 0x2 – test registers 0x3 – test FLASH 0x4 – test DDR(s) 0x5 – test FPGA interconnect To issue a command the MCU must write one of the above values to this register. The MCU can then poll this register to check if the test is done (register will return all zeros when finished)
Status Register	0x8C00_0008	0x8C00_0008	R	Read-only register - Status of commands and bus control: Bits 16-18 must all be zero for MCU to issue any commands to External Host Command Register. Status results can be read after command register is cleared. The decode for the test results is as follows: Bit 0 – overall pass/fail (pass = 1, fail = 0) Bit 1 – register test pass/fail Bit 2 – flash test pass/fail Bit 3 – ddr test pass/fail Bit 4 – interconnect test pass/fail
Existing FPGA Register	0x8C00_0018	0x8C00_0018	R/W	Contains information on what FPGAs are stuffed on the DN6000k10 as well as what type of FPGAs they are. The register has the following format: Bit 0 – 1 if FPGA A is stuffed, 0 otherwise Bit 1 – 1 if FPGA B is stuffed, 0 otherwise Bit 2 – 1 if FPGA C is stuffed, 0 otherwise Bit 2 – 1 if FPGA D is stuffed, 0 otherwise Bit 2 – 1 if FPGA E is stuffed, 0 otherwise Bit 2 – 1 if FPGA F is stuffed, 0 otherwise Bit 2 – 1 if FPGA G is stuffed, 0 otherwise Bit 7 – 1 if FPGA H is stuffed, 0 otherwise Bit 8 – 1 if FPGA I is stuffed, 0 otherwise Bit 9 – 1 if FPGA A is 2vp100, 0 if 2vp70 Bit 10 – 1 if FPGA B is 2vp100, 0 if 2vp70 Bit 11 – 1 if FPGA C is 2vp100, 0 if 2vp70 Bit 12 – 1 if FPGA D is 2vp100, 0 if 2vp70 Bit 13 – 1 if FPGA E is 2vp100, 0 if 2vp70 Bit 14 – 1 if FPGA F is 2vp100, 0 if 2vp70 Bit 15 – 1 if FPGA G is 2vp100, 0 if 2vp70 Bit 16 – 1 if FPGA H is 2vp100, 0 if 2vp70 Bit 17 – 1 if FPGA I is 2vp100, 0 if 2vp70